UNIVERSIDAD MIGUEL HERNANDEZ

Departamento de Ingeniería de Sistemas Industriales

RECONOCIMIENTO Y BÚSQUEDA SELECTIVA DE OBJETOS EN ENTORNOS NO ESTRUCTURADOS MEDIANTE APARIENCIA MULTICANAL Y MULTIESCALA

TESIS DOCTORAL

por

María Asunción Vicente Ripoll

Ingeniera en Electrónica

Elx, Marzo 2006

UNIVERSIDAD MIGUEL HERNANDEZ

Departamento de Ingeniería de Sistemas Industriales

RECONOCIMIENTO Y BÚSQUEDA SELECTIVA DE OBJETOS EN ENTORNOS NO ESTRUCTURADOS MEDIANTE APARIENCIA MULTICANAL Y MULTIESCALA

María Asunción Vicente Ripoll

Director: Dr. Óscar Reinoso García

Elx, Marzo 2006

TRIBUNAL

Presidente:

Vocal:

Vocal:

Vocal:

Vocal secretario:

Calificación de la tesis: .....

AUTORIZACIÓN DE PRESENTACIÓN
DE TESIS DOCTORAL

Director: Óscar Reinoso García

Título de la tesis:
Reconocimiento y búsqueda selectiva de objetos en entornos no estructurados mediante apariencia multicanal y multiescala.

Autora: María Asunción Vicente Ripoll

Departamento: Ingeniería de Sistemas Industriales

Universidad Miguel Hernández

El director de la tesis reseñada autoriza su presentación de acuerdo con lo dispuesto en el artículo 8º del R.D. 778/1998 de 30 de abril por el que se regula el tercer ciclo de estudios universitarios.

Elx, a .....

Firmado: Óscar Reinoso García

DEPARTAMENTO
DE INGENIERÍA DE SISTEMAS INDUSTRIALES

Óscar Reinoso García, Profesor Titular de Universidad y Director del Departamento de Ingeniería de Sistemas Industriales de la Universidad Miguel Hernández

Certifica

Que el trabajo realizado por Dª María Asunción Vicente Ripoll titulado *Reconocimiento y búsqueda selectiva de objetos en entornos no estructurados mediante apariencia multicanal y multiescala*, ha sido dirigido por el Dr. D. Óscar Reinoso García y realizado en el Departamento de Ingeniería de Sistemas Industriales, se encuentra en condiciones de ser leído y defendido como Tesis Doctoral ante el correspondiente tribunal en la Universidad Miguel Hernández.

Lo que firmo para los efectos oportunos en Elche, a .....

Fdo. Óscar Reinoso García
Director del Dpto. de Ingeniería de Sistemas Industriales

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS INDUSTRIALES
Campus de Elche. Avenida de la Universidad s/n - 03202 ELCHE
Tlf: 96 665 89 80 - Fax 96 665 98 79

Agradecimientos

Ante todo, quiero dar las gracias al director de esta tesis Dr. D. Oscar Reinoso, por el apoyo prestado durante este prolongado periodo, por sus consejos y también por sus críticas.

La tesis se ha realizado en el seno del Departamento de Ingeniería de Sistemas Industriales de la Universidad Miguel Hernández, y deseo también dar las gracias a todos sus miembros, particularmente a aquellos que han contribuido y siguen contribuyendo a hacer del Departamento un lugar agradable donde trabajar. Especialmente a Arturo, Luis, Luis Miguel, Rafa y Ramón, con los que he compartido muchos buenos momentos dentro y fuera del trabajo; a José Antonio y David, a Miguel Ángel, Emilio, Ramón y al resto de los mecánicos, por su amistad;a Maria José, Laura y Carolina por poner un poco de orden y femineidad en este departamento tan arcaicamente masculino.

Fuera del Departamento, pero también compañeros, quiero agradecer el apoyo de mis amigas: Geli, a quien debo el formato LaTeXde esta tesis; a Miguel, por su comprensión metafísica; y a Susana y Antonia, quienes me guiaron en el buen camino... Así como a mi gran amigo y compañero de fátigas *reconociendo objetos,* Pablo.

Algunos de los desarrollos de la tesis se han realizado en la Universidad de Helsinki (*Computer Science Department, Neuroinformatics*), y quiero también agradecer a Patrik y Aapo la grata acogida que me ofrecieron en su laboratorio, su ayuda imprescindible en la comprensión del método ICA para su aplicación en las tareas de reconocimiento de objetos, y proporcionarme ingentes cantidades de información acerca del nuevo método NMFSC. También quiero recordar a algunos compañeros del laboratorio que hicieron aún más fácil y enriquecedora la estancia en Helsinki: Juan Carlos, EviMaria, Michael, Ydo, Aristides, Panayiotis y Jarmo.

Más lejos del trabajo, quiero también dar las gracias a mis compañeras de inglés que tan buenos momentos me han hecho pasar, Lorraine, Carrie, Charlotte, Miranda y Samantha, así como a mis queridas amigas Lourdes, Raquel H., Lidia, Ana, M. Luisa, e Isabel, un ejemplo a seguir, si es que estás lo suficientemente entrenado.

Tambien quiero recordar a mis padres, Gloria y Manolo, y mis hermanos, Marga y Manolo, por su apoyo infinito y su buen consejo, así como mis queridísimas sobrinitas Isabel y Lucía. Por último, tengo que decir que podría escribir toda una tesis expresando mi gratitud a César, sin él, este documento y parte de mis artículos no existirían, puesto que la mayoría de las aportaciones de la presente tesis se han realizado conjuntamente con él; gracias por todas tus ideas, sugerencias, críticas y revisiones, además de tener la paciencia de soportarme y animarme a todas horas y todos los días mientras sufría la escritura de este documento.

Resumen

La presente tesis se centra en el reconocimiento de objetos de forma libre mediante visión artificial. Específicamente, está orientada hacia el reconocimiento de objetos basado en la apariencia, esto es, haciendo uso de toda la información del objeto presente en sus imágenes más representativas.

Los métodos de reconocimiento de objetos basados en la apariencia hacen uso de técnicas de extracción de características para comprimir los modelos de apariencia de los objetos a reconocer. Uno de objetivos de esta tesis es aclarar una serie de contradicciones existentes en la actualidad, relativas a qué método de extracción de características lineal es más idóneo en ciertas circunstancias de trabajo. Concretamente, se pretende aportar algo de luz a la falta de congruencia entre los resultados obtenidos mediante el análisis de componentes independientes (ICA) y los obtenidos con el método clásico de análisis de componentes principales (PCA).

Otra área de trabajo de la tesis se centra en el análisis de nuevos métodos de extracción de características, como es el caso de las proyecciones aleatorias (RP) o la factorización en matrices no negativas (NMF). En esta tesis se señala bajo qué condiciones estos nuevos métodos pueden mejorar las propiedades de técnicas clásicas como el análisis de componentes principales o el análisis discriminante lineal (LDA). De este modo, se propone una nueva arquitectura (el subespacio objeto modificado) para la construcción de los modelos de apariencia de los objetos, que permite mejorar los resultados de reconocimiento utilizando la factorización en matrices no negativas.

El estudio detallado de todas estas técnicas de extracción de características concluye con el desarrollo de un novedoso sistema de detección de objetos de forma libre en entornos no estructurados, basado en el uso de múltiples canales de información (apariencia multicanal) y que permite la detección del objetos en escalas diversas (multiescala). El nuevo método de detección propuesto presenta las siguientes características: combina dos etapas de búsqueda en paralelo, hace uso de dos métodos de extracción de características (análisis de componentes principales y factorización de matrices no negativas con restricciones de raleza (NMFSC)), realiza un ajuste automático de la influencia de cada uno de los canales de información en función del objeto a detectar y genera, a partir de la información de entrenamiento del objeto de interés, un umbral automático de detección.

El sistema de búsqueda de objetos propuesto se lleva a la práctica sobre un entorno real, presentándose además un análisis detallado de la capacidad de detección de dicho sistema.

Abstract

This thesis is focused on free-form object recognition using computer vision, specifically, on those tecniques based on appearance.

The appearance-based object recognition approaches use feature extraction techniques in order to compress the representative models of the target objects. Recently, a number of empirical studies have claimed that Independent Component Analysis (ICA) outperforms Principal Component Analysis (PCA) as a feature extraction method in classification systems, although others state that both approaches perform equally, and other researchs claim that PCA is superior to ICA. One of the goals of this thesis is to clarify those seemingly contradictory results and to clarify under which circumstances ICA may outperform PCA.

Other feature extraction techniques like Random Projection (RP) and Non-negative Matrix Factorization (NMF) are also studied. In this thesis it is shown that these new methods can outperform classical approaches like PCA or Linear Discriminant Analysis (LDA). A novel architecture is proposed, namely, the modified object subspace, which allows to obtain the maximum performance in NMF approaches.

Finally, the previous results are applied in a novel free-form object detection system. Such system can work in unstructured environments, and perfoms both a multiscale and a multichannel search. These searches are performed in parallel using different feature extraction methods (PCA and NMFSC). Besides, the channels are weighted automatically, and the detection threshold is also automatically adjusted from training images.

Some results are presented where the good behaviour of the system on a real environment is shown.

Resum

La present tesi es centra en el reconeixement d'objectes de forma lliure mitjançant la visió artificial. Específicament, està orientada cap al reconeixement d'objectes basat en l'aparença, açò és, fent ús de tota la informació de l'objecte present en les seues imatges més representatives.

Els mètodes de reconeixement d'objectes basats en l'aparença fan ús de tècniques d'extracció de característiques per a comprimir els models d'aparença dels objectes a reconéixer. Un dels objectius d'aquesta tesi és aclarir una sèrie de contradiccions existents en l'actualitat, relatives a quin mètode d'extracció de característiques lineal és mes idoni en certes circumstàncies de treball. Concretament, es pretén aportar llum al fet que a l'utilitzar l'anàlisi de components independents no sempre s'obtenen uns resultats equivalents als generats pel mètode clàssic d'anàlisi de components principals.

Una altra àrea de treball de la tesi es centra en l'anàlisi de nous mètodes d'extracció de característiques, com és el cas de les projeccions aleatòries o la factorització en matrius en no negatives. En esta tesi s'assenyalen davall què condicions estos nous mètodes poden millorar les propietats de tècniques clàssiques com l'anàlisi de components principals o l'anàlisi discriminant lineal. D'aquesta manera, es proposa una nova arquitectura (el subespai objecte modificat) per a la construcció dels models d'aparença dels objectes que permet millorar els resultats de reconeixement utilitzant la factorització en matrius no negatives.

L'estudi detallat de totes aquestes tècniques d'extracció de característiques culmina en el desenvolupament d'un nou sistema de detecció d'objectes de forma lliure en entorns no estructurats, basat en l'ús de múltiples canals d'informació (aparença multicanal) i que permet la detecció d'objectes en escales diverses (multiescala). El nou mètode de detecció proposat presenta les següents característiques: combina dues etapes de recerca en paral·lel, fa ús de dos mètodes d'extracció de característiques (l'anàlisi de components principals i la factorització de matrius no negatives amb restriccions d'atapeïment), realitza un ajust automàtic de la influència de cadascun dels canals d'informació en funció de l'objecte a detectar i genera, a partir de la informació d'entrenament de l'objecte d'interés, un llindar automàtic de detecció.

El sistema de recerca d'objectes proposat es materialitzta sobre un entorn real, presentant-se a més un anàlisi detallat de la capacitat de detecció del sistema dissenyat.

Índice general

Índice general **XVII**

Índice de figuras **XXIV**

Índice de tablas **XXXVI**

1. Introducción **1**
1.1. Motivación y objetivos 1
1.2. Marco de la tesis 4
1.3. Resumen de las aportaciones realizadas 5
1.4. Estructura de la tesis 6

2. Reconocimiento de objetos mediante visión por computador **7**
2.1. Teorías Computacionales de Reconocimiento de Objetos 9
2.1.1. Métodos basados en modelo 11
2.1.1.1. Descomposiciones estructurales 11
2.1.1.2. Restricciones geométricas 15
2.1.2. Métodos basados en apariencia 21
2.2. Estructura de los sistemas de reconocimiento de objetos basados en la apariencia visual 21
2.3. Propuestas previas de sistemas de reconocimiento basados en la apariencia 25
2.3.1. Sistemas de reconocimiento basados en PCA 25
2.3.2. Sistemas de reconocimiento basados en LDA 28
2.3.3. Sistemas basados en ICA 28
2.3.4. Sistemas basados en otras transformaciones 30
2.4. Objetos de forma libre y sus librerías 32
2.5. Conclusiones 37

3. Técnicas de extracción de características **39**
3.1. Descripción de los procesos básicos que constituyen los sistemas de reconocimiento de patrones 41
3.1.1. Módulo de adquisición de datos 41
3.1.2. Módulo de selección y extracción de características 42

3.1.2.1. Selección de características 44
3.1.2.2. Extracción de características 45
3.1.3. Módulo de clasificación: aprendizaje y entrenamiento 47
3.1.3.1. Conceptos generales 47
3.1.3.2. Clasificadores supervisados 50
3.1.3.3. Clasificadores no supervisados 52
3.1.4. Redes neuronales 54
3.2. Análisis de Componentes Principales, PCA 55
3.2.1. PCA por maximización de la varianza 55
3.2.2. Propiedades de PCA 59
3.2.3. PCA por minimización del error cuadrático medio 63
3.2.4. Algunas consideraciones sobre PCA 66
3.3. Análisis de Fisher, LDA 68
3.4. Análisis de Componentes Independientes, ICA 72
3.4.1. Modelo ICA 72
3.4.2. Estimación del modelo ICA 72
3.4.3. Codificación *sparse* 74
3.4.4. Búsqueda de proyecciones no-gaussianas 75
3.4.5. El algoritmo FastICA 78
3.4.6. Funciones utilizadas en la estimación del modelo ICA 79
3.4.7. Areas de aplicación de ICA 81
3.5. Proyecciones aleatorias, RP 84
3.6. Factorización de Matrices No Negativas, NMF 86
3.6.1. El modelo NMF 86
3.6.2. NMF con restricciones de codificacion *sparse* 89
3.6.3. Algunas consideraciones sobre NMFSC 90

4. Arquitectura de reconocimiento de objetos propuesta 93
4.1. Subespacios Universal y Objeto 95
4.2. Arquitectura propuesta: Subespacio Objeto Modificado 100
4.3. Estudio del funcionamiento de la arquitectura propuesta con la librería COIL-100 114
4.3.1. Resultados con el subespacio universal PCA 114
4.3.2. Uso de más vistas para subespacio universal 125
4.3.3. Resultados con los subespacios objeto modificados PCA 127
4.4. Estudio del funcionamiento de la arquitectura propuesta con la librería de objetos UMH-15 134
4.4.1. Resultados con el subespacio universal PCA 136
4.4.2. Uso de más vistas para subespacio universal 138
4.4.3. Resultados con los subespacios objeto modificados PCA 139
4.5. Conclusiones 141

5. Métodos de extracción de características propuestos 143
5.1. ICA vs. PCA . . . 145
5.1.1. Capacidad de separación de clases de cada uno de los métodos . . 145
5.1.1.1. PCA, SVD y *whitening* . . . 145
5.1.1.2. ICA . . . 147
5.1.1.3. Diferentes arquitecturas para la descomposición ICA: ¿independencia en A o en S? . . . 151
5.1.1.4. Clasificadores invariantes ante rotación . . . 153
5.1.1.5. Selección de subconjuntos de componentes . . . 154
5.1.2. Posibles fuentes de error en la comparación de PCA e ICA . . . 159
5.1.2.1. PCA vs. whitening . . . 161
5.1.2.2. Influencia de la normalización . . . 172
5.1.3. Conclusiones . . . 183
5.2. RP vs. PCA . . . 185
5.2.1. Ventajas de aplicar RP al reconocimiento de objetos . . . 185
5.2.2. Resultados experimentales . . . 188
5.3. NMFSC vs. PCA . . . 191
5.3.1. Preprocesado de los datos . . . 191
5.3.2. Estudio del funcionamiento de NMFSC utilizando la arquitectura de subespacio objeto modificado . . . 200
5.3.2.1. Comparativa de los canales propuestos . . . 200
5.3.2.2. Comparativa PCA vs. NMFSC . . . 203

6. Aplicación a la detección de objetos en escenas no estructuradas 213
6.1. Tipo de aplicación: detector de objetos basado en apariencia . . . 215
6.1.1. Descripción de la búsqueda selectiva basada en apariencia multicanal y multiescala . . . 220
6.1.1.1. Procesos *off-line* . . . 220
6.1.1.2. Procesos *on-line* . . . 225
6.1.1.3. Combinación de las etapas multiescala y multicanal . . . 231
6.1.2. ¿Por qué utilizar varios canales de información? . . . 232
6.1.3. ¿Por qué utilizar fragmentos de las imágenes de entrenamiento? . . . 233
6.1.4. ¿Por qué utilizar dos tipos de compresión? . . . 241
6.2. Canales de información utilizados en la búsqueda selectiva multicanal y multiescala . . . 246
6.2.1. Canales de apariencia . . . 246
6.2.1.1. RGB y rgb . . . 246
6.2.1.2. HSV y HSI . . . 249
6.2.1.3. $\mathbf{l_1 l_2 l_3}$ y $\mathbf{c_1 c_2 c_3}$ . . . 251
6.2.2. Canales de histogramas . . . 253
6.2.3. Canales de forma . . . 256
6.2.4. Canal ON-OFF . . . 258

6.3. Ajuste automático de la influencia de cada canal 259
6.3.1. Fiabilidad de cada canal de información 260
6.3.2. Propuesta de aprendizaje de pesos mediante entrenamiento 260
6.4. Umbral automático de reconocimiento 269
6.4.1. Necesidad de establecer un umbral 269
6.4.2. Establecimiento del umbral a través de la distribución espacial de los modelos de apariencia 270
6.4.3. Establecimiento del umbral mediante aprendizaje con escenas etiquetadas 273
6.4.4. Uso de los umbrales 281

7. Resultados experimentales **283**
7.1. Entorno de pruebas utilizado 284
7.1.1. Captura de imágenes con webcam 284
7.1.2. Fragmentos de las imágenes 285
7.1.3. Base de datos de objetos a reconocer 286
7.1.4. Condiciones de captura de las escenas para la detección 286
7.1.5. Condiciones de captura de las imágenes para la generación de los modelos de cada objeto 290
7.2. Aplicaciones desarrolladas 291
7.2.1. Entorno de entrenamiento 292
7.2.2. Entorno de experimentación para comprobar la influencia de cada canal y el efecto de los umbrales 295
7.2.3. Entorno de funcionamiento automático 305
7.3. Capacidad de detección de los distintos canales. 307
7.4. Capacidad de detección del canal LoG_{ON-OFF} y de la combinación de los procedimientos multicanal y multiescala 317

8. Conclusiones **319**
8.1. Objetivos alcanzados 319
8.2. Aportaciones de la tesis 320
8.3. Aplicabilidad práctica 321
8.4. Trabajos futuros 323

A. Notación **325**

B. Algoritmo NMFSC **331**
B.1. Algoritmo NMFSC 331
B.2. Operador de proyección 332

C. Modelos y distribuciones espaciales del objeto *caja amarilla* **333**
C.1. Modelos de apariencia del objeto *caja amarilla* 333

C.2. Distribuciones espaciales de los modelos de apariencia del objeto *caja amarilla* 341

Bibliografía **349**

Índice de figuras

1.1. Detalle del robot móvil B21r en un laboratorio de la universidad. Si se desea que el robot interactúe con los humanos ha de ser capaz de detectar caras y reconocerlas bajo distintos tipos de iluminación, fondos complejos y diferentes distancias respecto de su posición. 2

2.1. En la figura de izqda. se muestra un conjunto de 5 *geones* y en la dcha., unos objetos representados según la teoría RBC. 12

2.2. Inconvenientes de las representaciones estructurales. Fig. 2.2(a): la inclusión de detalles métricos reduce la habilidad para reconocer nuevos objetos. Fig. 2.2(b): Una imagen de un carrito de *hot-dogs* y su objecto en representación RBC según Biederman [19]. Actualmente es imposible segmentar en la imagen original de niveles de gris las primitivas que generan los geones. Fig. 2.2(c): Incluso en las tareas de reconocimiento más simples como el reconocimiento de caracteres, derivar una descripción estructural única es problemático. 14

2.3. Objeto a reconocer: esquinas numeradas con letras. La cara más significativa del objeto es la formada por las esquinas GHIJKL al disponer de 6 puntos-esquinas 16

2.4. Invariantes obtenidos a partir de las esquinas señaladas de la figura 2.5 . . 17

2.5. Diferentes vistas del objeto con las esquinas significativas señaladas. . . . 18

2.6. Detalles del reconocimiento de objetos basado en el uso de invariantes de dos cónicas no coplanares[107]: a)imágenes estereoscópicas del objeto a reconocer, b)representación esquemática del invariante utilizado; c)gráfico con las cónicas detectadas; d)cálculo de los modelos que identifican al objeto, la cuarta fila es el valor del invariante proyectivo utilizado en distintas configuraciones del objeto. 20

2.7. En el *espacio imagen* las imágenes de los objetos son puntos pertenecientes a un espacio multidimensional $M = f \times c$ 22

2.8. Una colección de N imágenes es equivalente a N puntos en el espacio imagen 23

2.9. La figura 2.9(a) muestra el conjunto de imágenes originales de caras humanas de la librería CBCL [2] y la figura 2.9(b) los vectores NMF obtenidos con el algoritmo de Lee y Seung[96]. 31
2.10. Varias vistas de imágenes de rango de *objetos de forma libre* artificiales presentados por Dorai y Jain[46]. 32
2.11. Dispositivo compuesto por una mesa giratoria y una cámara para la adquisición de las imágenes de los objetos de la librería COIL-100. 33
2.12. Conjunto de los objetos que componen la librería de imágenes de objetos de forma libre COIL-100 33
2.13. Ejemplo de las imágenes de un objeto de la librería ALOI visto bajo 72 ángulos distintos (paso de 5°). 34
2.14. Ejemplo de las imágenes de un objeto de la librería ALOI visto bajo 24 direcciones de iluminación diferentes. 34
2.15. Ejemplo de las imágenes de un objeto de la librería ALOI visto bajo 12 colores de iluminación (temperaturas) diferentes. 35
2.16. Algunos ejemplos de escenas complejas con objetos de la librería SOIL-47. 35
2.17. Esfera geodésica. 36
2.18. Conjunto de los objetos que componen la librería de imágenes de objetos de forma libre ETH-80 36

3.1. Esquema modular de un sistema de reconocimiento de patrones. 42
3.2. Esquema modular de un sistema de reconocimiento de objetos mediante imágenes bidimensionales. 43
3.3. Diversos enfoques en el reconocimiento estadístico de patrones. 51
3.4. Ejemplo ilustrativo de la transformación PCA: datos originales (puntos negros), datos centrados (puntos rojos), datos proyectados en las direcciones de los vectores propios (puntos azules), los ejes principales están en azul. 59
3.5. Esta figura ilustra el proceso de PCA. Arriba, la primera fila recoge las imágenes en color de 9 conocidos pilotos de motociclismo, y son las imagenes de entrada utilizadas para generar el eigespacio. En la segunda fila, se muestran los ocho vectores propios obtenidos y la imagen media. En el resto de filas se muestran las reconstrucciones junto con su error de reconstrucción. 65
3.6. Ejemplos de conjuntos de datos diferentes cuyas componentes principales son iguales. 66
3.7. Transformaciones LDA y PCA sobre un conjunto de puntos compuesto por dos clases. Se aprecia como al reducir los puntos originales a una sola dimensión, la transformación LDA consigue separar las dos clases, mientras que la primera componente principal no lo consigue totalmente. 70

3.8. En este ejemplo, los resultados de clasificación de PCA (utilizando 1 solo vector propio) son superiores a los de LDA. [111] 71
3.9. Se muestran cuatro ejemplos de vectores con distinto nivel de *sparsity*. Los niveles de *sparseness* que se muestran son 0.1, 0.4, 0.7 y 0.9. Cada barra denota el valor de cada uno de los elementos del vector. Si el nivel de *sparsity* es bajo, todos los elementos son igualmente activos, sin embargo, para valores de *sparsity* elevados, la mayoría de los coeficientes son cero y sólo unos pocos tienen valores significativos. 74
3.10. Ejemplos de distribuciones con distinto grado de no-gaussianidad. Si se utiliza la kurtosis como medida de no-gaussianidad, esta es positiva para las distribuciones supergaussianas, negativa para las subgaussianas y cero para las normales 77
3.11. ICA en Neuroinformática. 83
3.12. Algunas imágenes de la base de datos de caras ORL. 88
3.13. Características obtenidas con la base de datos de caras ORL utilizando el algoritmo clásico de NMF. Se obtienen características globales y no locales de las imágenes de entrenamiento. 88
3.14. Características obtenidas con la base de datos de caras ORL utilizando el algoritmo NMFSC para distintos valores de sW 89
3.15. Aplicación de NMF y NMFSC a un mismo conjunto de imágenes (objeto nº 29 de la COIL-100). Se aprecia cómo solamente NMFSC es capaz de extraer o resaltar características locales de las imágenes. 91

4.1. Ejemplo de un conjunto de imágenes universal con algunos objetos (1, 29, 68 y 75) de la librería COIL-100. 96
4.2. Ejemplo de subespacio universal: los 20 primeros vectores PCA obtenidos a partir de las 72 imágenes de cada uno de los cuatro objetos mostrados en la figura 4.1 97
4.3. Proyecciones ($\mathbf{P}$) del conjunto de imágenes universal de 4 objetos de la COIL-100 (72 imágenes por cada objeto) en un subespacio universal PCA de 2 dimensiones. 99
4.4. Ejemplo de subespacio objeto: trayectoria en el subespacio objeto PCA del objeto nº 14 de la librería COIL-100, utilizando sólo 3 dimensiones . 99
4.5. Imagenes de los objetos 1, 29 y 68 de la librería COIL-100. Conjuntos de imágenes de entrenamiento: (a) Objeto 1; (c) Objeto 29; (e) Objeto 68. Conjunto de imágenes de test: (b) Objeto 1; (d) Objeto 29; (f) Objeto 68. . 103
4.6. Representación bidimensional de los subespacios objeto PCA en el canal I de los objetos 1, 29 y 68 de la librería COIL-100. 104
4.7. Proyección de las imágenes de test del objeto 1 sobre cada uno de los subespacios objeto. 105
4.8. Proyección de las imágenes de test del objeto 29 sobre cada uno de los subespacios objeto. 106

4.9. Proyección de las imágenes de test del objeto 68 sobre cada uno de los subespacios objeto. 107
4.10. Subespacios objeto modificados 108
4.11. Subespacios objeto modificados 109
4.12. Librería COIL-20 116
4.13. Tasa de éxito en el reconocimiento con subespacio universal PCA para las imagenes de la COIL-20 en los tres canales, con distancia de Mahalanobis y entrenamiento tipo *frontal*. Los objetos considerados simétricos tienen una tasa del 100 % 118
4.14. Tasa de éxito en el reconocimiento con subespacio universal PCA para las imagenes de la COIL-20 en los tres canales, con la norma L2 y entrenamiento tipo *frontal*. Los objetos considerados simétricos tienen una tasa del 100 % 119
4.15. Entrenamiento *frontal* vs *aleatorio* para el canal I 120
4.16. Entrenamiento *frontal* vs *aleatorio* para el canal binario 121
4.17. Entrenamiento *frontal* vs *aleatorio* para el canal de bordes 122
4.18. Oclusiones 123
4.19. Tasa de éxito en el reconocimiento para canal I con oclusion del 25 %, los objetos simétricos se reconocen al 100 % 124
4.20. Objetos de la COIL-20 en grupos según la variablidad de sus vistas 127
4.21. Tasa de éxito en el reconocimiento para canal I, con la arquitectura subespacio objeto modificado PCA y distancia de Mahalanobis (36 vistas de entrenamiento) 130
4.22. Tasa de éxito en el reconocimiento para canal I, con la arquitectura subespacio objeto modificado PCA y norma L2 (36 vistas de entrenamiento) 131
4.23. Librería UMH-15 134
4.24. Objetos 8 y 10 de la librería UMH-15 135
4.25. Objetos simétricos de la librería UMH-15 137

5.1. Dos ejemplos artificiales: un conjunto de datos subgaussiano (1ª fila) y un conjunto supergaussiano (2ª fila) (a), ambos transformados por PCA (b), *whitening* (c) y utilizando el modelo ICA a través del algoritmo *FastICA* (d). La última figura de cada fila (e) muestra el conjunto de datos original con las direcciones de PCA e ICA. 148
5.2. Dos ejemplos artificiales: un conjunto de datos subgaussiano (1ª fila) (a) y un conjunto supergaussiano (2ª fila), ambos transformados utilizando el modelo ICA a través del algoritmo *Extended Infomax* (b). Los datos blanqueados, se han copiado de la Fig.5.1 para su comparación. La última figura de cada fila (d) muestra el conjunto de datos original con las direcciones de PCA e ICA. 149
5.3. Diferentes arquitecturas para la descomposición ICA de un conjunto de imágenes: ¿queremos la independencia en $\mathbf{A}$ o en $\mathbf{S}$? 151

5.4. Diferentes arquitecturas para la descomposición ICA de un conjunto de imágenes: ¿queremos la independencia en $\mathbf{A}$ o en $\mathbf{S}$? 152
5.5. La *mejor* selección de características puede ser diferente según la distribución de las clases en el conjunto de datos. En la figura 5.5(a) las clases están perfectamente separadas utilizando sólo la proyección sobre la dirección ICA definida por el vector $\mathbf{w}_1$, mientras que en la figura 5.5(b), las clases están mejor separadas utilizando la dirección del eigenvector $\mathbf{u}_2$ 154
5.6. Resultados de clasificación para los datos de ICA y PCA blanqueados utilizando subconjuntos de características. 156
5.7. Subespacios universales: tasa de éxito en el reconocimiento para PCA y *whitening* utilizando solo 10 componentes. 162
5.8. Subespacios universales: tasa de éxito en el reconocimiento para PCA y *whitening* en relación con el número de componentes del subespacio. . . . 162
5.9. (a) Objetos simétricos con PCA: $\{2, 4, 5, 18, 24, 25, 26, 30, 32, 33, 34, 35, 47, 50, 56, 58, 61, 64, 70, 72, 83, 86, 87, 88, 94, 95\}$; (b) Objetos simétricos con *whitening*: $\{5, 18, 24, 25, 26, 30, 32, 34, 35, 47, 50, 56, 58, 61, 70, 72, 83, 86, 87, 94, 95\}$ 164
5.10. (a) 4 objetos *simétricos* y 4 *no-simétricos* de la COIL-100;(b) error para PCA y *whitening* para los 4 objetos *no-simétricos*; (c) error para PCA y *whitening* para los 4 objetos *simétricos*. 165
5.11. Representación bidimensional del índice de separabilidad trivial. 166
5.12. Indices de separabilidad, α_t (a), y β_t (b) para PCA y *whitening* con los 20 primeros objetos de la COIL-100 y arquitectura universal. 167
5.13. Índice de separabilidad para los subespacios objeto modificados PCA y *whitening* para la librería de objetos completa en el canal I. Los subespacios objeto se han obtenido utilizando solo 8 imágenes de entrenamiento por objeto. 168
5.14. Índice de separabilidad para los subespacios objeto modificados de los 20 primeros objetos de la COIL100 (canal I) utilizando subespacios de diferentes dimensiones (desde 2 a 24 componentes): (a) PCA, (b) *whitening*. 170
5.15. Índice de separabilidad para los subespacios objeto modificados de los 20 primeros objetos de la COIL100 (canal histograma RGB) utilizando subespacios de diferentes dimensiones (desde 2 a 24 componentes): (a) PCA, (b) *whitening*. 171
5.16. Experimento de normalización 173
5.17. PCA Universal con 20 objetos. 174
5.18. PCA Universal, influencia de la normalización de los datos. 175
5.19. Imagenes de entrenamiento y test para el coche, utilizadas en los experimentos de capacidad de reconocimientoo con ruido salt & pepper (ruido sólo en imágenes de test). 177
5.20. PCA Universal, Ruido Gaussiano 179
5.21. PCA Universal, Ruido Salt & Pepper. 180

5.22. PCA Objeto Modificado, canal I, influencia de la normalización de los datos. 181
5.23. Gráfico extraído de Fortuna *et al.* [61], queda patente que en el caso de utilizar todas las dimensiones, los resultados de ambos métodos coinciden. 182
5.24. Tiempos de cómputo *off-line* de distintos métodos de extracción de características. 186
5.25. Un subconjunto de 20 objetos de la librería COIL-100: los 20 objetos más similares entre sí en apariencia y por tanto, los más complejos de clasificar correctamente. 188
5.26. Tiempos de cómputo *off-line* de distintos métodos de extracción de características. 189
5.27. Imágenes en los canales ON-OFF obtenidas a partir de la imagen *Lenna* transformada por los operadores de segundo orden L y LoG. 195
5.28. Imágenes de las diferencias entre los canales ON-OFF y sus fuentes, L y LoG. 196
5.29. Vectores de NMFSC, $S_w = 0{,}7 S_h = 0$, obtenidos de un mismo conjunto de imágenes (objeto nº 10 de la COIL-100) preprocesadas mediante el canal ON-OFF sobre las imágenes en niveles de grises (a); y mediante el canal ON-OFF sobre las imágenes filtradas por el operador laplaciano (b). 198
5.30. Vectores de NMFSC para los objetos 7 y 81 de la COIL-100, con tres grados diferentes de *sparseness*, $S_W = 0{,}4, 0{,}7$ y $0{,}75$, partiendo de las imágenes de bordes (operador laplaciano). 199
5.31. Comparativa del funcionamiento de los canales propuestos con NMFSC. . 201
5.32. Comparativa con los canales ON-OFF para $S_w = 0{,}4$. 202
5.33. Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal L_{ON-OFF}. 204
5.34. Comparativa con subespacios objeto modificados en el canal laplaciano (*ON-OFF-L*). 205
5.35. Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal LoG_{ON-OFF}. 207
5.36. Comparativa con subespacios objeto modificados en el canal LoG_{ON-OFF}. 208
5.37. Imágenes de dos de los objetos de la librería COIL-100 (en el canal L) utilizados durante los experimentos con oclusiones. 209
5.38. Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal L_{ON-OFF}y con presencia de oclusiones en las imágenes de test. 210
5.39. Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal LoG_{ON-OFF} y con presencia de oclusiones en las imágenes de test. 211
5.40. Comparativa con subespacios objeto modificados para los dos canales en presencia de oclusiones. 212

6.1. Ejemplos de detectores de objetos genéricos. 216
6.2. Tarea a realizar por el detector de objetos: localizar regiones en una escena que muestren al objeto u objetos de interés. En esta figura se muestra una escena compleja con múltiples objetos de la librería COIL-100. 216
6.3. (a)División de la escena de la figura anterior en ventanas de tamaño 80×130 con un paso de 20 píxeles. La escena original tiene un tamaño de 474×472 píxeles. (b)Un conjunto de 7 vistas representativas del objeto 3 de la librería COIL100. 219
6.4. Estructura general del algoritmo de detección de objetos propuesto. . . . 222
6.5. Estructura de la detección multicanal PCA. 223
6.6. Estructura de la detección multiescala NMF. 224
6.7. Los fragmentos obtenidos a partir de las imágenes de entrenamiento del objeto se comprimen utilizando PCA. 225
6.8. La división de la escena en ventanas depende del tamaño de la ventana y el paso utilizado para recorrer toda la imagen. El tamaño y paso de la ventana está expresados en número de píxeles. 227
6.9. Las matrices de repetición de píxeles indican las veces que un mismo píxel aparece en varias ventanas y cabe tenerlas en cuenta durante la construcción de los mapas de probabilidad de la escena. 228
6.10. Normalización de los mapas de distancia. 229
6.11. Resumen de los principales resultados del trabajo de Ullman *et al.* sobre el contenido de información en los fragmentos de imágenes de caras y coches. (a) Ejemplos de las imágenes de caras y coches utilizadas para generar los fragmentos. (b) Los mejores 8 fragmentos alrededor de una cara, en orden decreciente de información mutua desde arriba y en sentido contrario a las agujas del reloj. (c) Un conjunto de fragmentos de imágenes de coches. (d) Los 8 mejores fragmentos de caras con su valor de *merit*(△) y *weight*(□). Superioridad de las características de complejidad intermedia: *Merit* y *weight* de los fragmentos de caras (e) y coches (f) en función del tamaño del fragmento. 235
6.12. El uso de fragmentos facilita la programación del algoritmo de búsqueda: (a) imágenes completas de dos objetos; (b) el área que ocupan en la imagen; (c) fragmentos cuadrados y rectangulares de los objetos, que permiten el uso de ventanas regulares durante la búsqueda del objeto en la escena. 237
6.13. *Merit* de los fragmentos de la caja amarilla en función del tamaño del fragmento. 239
6.14. *Merit* promediado de los fragmentos de 10 objetos en función del tamaño del fragmento. 240
6.15. División de una escena en múltiples ventanas. Cada cuadrado de la ilustración equivale a un píxel. 242
6.16. Relación entre el número de ventanas y el paso del barrido. 243

6.17. Relación entre el número de ventanas y su tamaño. 243
6.18. La búsqueda multiescala consiste en el barrido de la escena varias veces con tamaños de ventanas distintos, o de modo equivalente, para un tamaño de ventana fijo, aplicar factores de escalado sobre la imagen original. . . 244
6.19. Espectro de luz visible. 247
6.20. Modelo de color RGB. 247
6.21. Modelo de color HSV: (a) representación sobre una rueda de color; (b) representación cónica. 250
6.22. Modelo de color HSI. 250
6.23. Diferencia entre los modelos de color HSV (izqda.) y HSI (dra). 250
6.24. Espacios de color utilizados la etapa de detección multicanal PCA. 252
6.25. Histogramas utilizados la etapa de detección multicanal PCA. 255
6.26. Imágenes de la *caja amarilla* en cada uno de los espacios de color utilizados la etapa de detección multicanal PCA. 256
6.27. Imágenes de bordes de la *caja amarilla* en cada uno de canales de color anteriores. 257
6.28. Escenas originales. 261
6.29. Escenas etiquetadas. 262
6.30. Ejemplo sobre el ajuste automático de pesos de los canales: (a) Escena utilizada para el cálculo de los pesos; (b) Escena etiquetada; (c)-(ñ) Canales de apariencia, junto con su correlación normalizada respecto de la escena etiquetada;(o) Pesos de cada canal, que en este caso, al utilizar sólo una escena de entrenamiento, son iguales a la correlación de cada canal. . 263
6.31. Transformación en los valores de los píxeles de los nuevos mapas de probabilidad $\tilde{\mathcal{M}}_i$. 265
6.32. Efecto del escalado no lineal en los mapas de probabilidad sobre la ponderación de los canales. 266
6.33. Evolución del vector de pesos al añadir imágenes etiquetadas de entrenamiento. 268
6.34. El mapa de probabilidades indica para cada píxel de la escena la probabilidad de que el objeto a detectar ocupe tal píxel. 269
6.35. Es necesario el establecimiento de un umbral para la correcta detección del objeto. 270
6.36. Las distancias existentes entre los fragmentos de entrenamiento se pueden representar en forma de histograma. 271
6.37. La distribución espacial del modelo de apariencia en un cierto canal puede utilizarse para determinar de modo automático un umbral de reconocimiento. 272
6.38. Escena original, escena etiquetada y mapa de probabilidad de la escena. . 273
6.39. (a) Al establecer un umbral de reconocimiento (0.57, en este ejemplo), es posible segmentar el objeto de la escena; (b) tipos de píxeles que aparecen en la escena umbralizada; (c) leyenda de los tipos de píxeles. 274

6.40. Variación de los porcentajes de error en función del umbral. 275
6.41. Histogramas de probabilidad de pertenencia al objeto. (a) Píxeles pertenecientes a la zona etiquetada como objeto. (b) Píxeles pertenecientes a la zona etiquetada como fondo. 276
6.42. Combinación de los histogramas de píxeles etiquetados como objeto y como fondo. 277
6.43. Distintas selecciones del umbral en función del porcentaje de falsos positivos y falsos negativos deseado. 277
6.44. Falsos positivos y falsos negativos en función del umbral. 279
6.45. Efecto de la variación de los porcentajes de error en función del umbral. . 280

7.1. Captura de imágenes de entrenamiento con una webcam. 284
7.2. Conjunto de imágenes de entrenamiento. 285
7.3. Conjunto de fragmentos de las imágenes de entrenamiento de la *caja amarilla*. 285
7.4. Obtención de los fragmentos de las imágenes de entrenamiento de la *caja amarilla* en los distintos espacios de color. 286
7.5. Algunos de los objetos utilizados durante las pruebas. 287
7.6. Ejemplos de escenas utilizadas para el entrenamiento y durante la detección. 288
7.7. Ejemplos de escenas utilizadas para el entrenamiento y durante la detección. 289
7.8. Un conjunto de imágenes de entrenamiento para uno de los objetos utilizados durante las pruebas. 290
7.9. Proceso de obtención de los fragmentos. 293
7.10. Creación de escenas etiquetadas. 294
7.11. Escena para la detección del objeto *la caja amarilla* 295
7.12. Entorno *visualiza mapas*: los 38 mapas de probabilidad de la etapa de búsqueda multicanal PCA obtenidos para la escena de la figura 7.11. . . . 296
7.13. Entorno *visualiza mapas*: los 38 mapas de probabilidad de la etapa de búsqueda multicanal PCA obtenidos para la escena de la figura 7.11 con realce $\gamma = 5$. 297
7.14. Entorno *visualiza mapas*: mapas de probabilidad de la etapa de búsqueda multiescala NMFSC de la escena de la figura7.11 (Los factores de escalado de la escena vienen expresados en tanto por uno). 298
7.15. Entorno *visualiza mapas*: mapas de probabilidad de la etapa de búsqueda multiescala NMFSC de otra escena compleja (se muestra en la esquina inferior derecha). Se aprecia como ciertas escalas producen resultados muy satisfactorios, mientras que en otras escalas el objeto desaparece completamente. (Los factores de escalado de la escena en este caso vienen expresados en tanto por cien). 299
7.16. Entorno *visualizador interactivo*: permite visualizar los mapas de probabilidad obtenidos durante las etapa de búsqueda y examinar el efecto de los pesos y el umbral aplicado. 300

7.17. Entorno *visualizador interactivo*. En este caso, el mapa de probabilidad global se ha obtenido sólo a partir de la ponderación de tres canales (Apariencia S y c_1, y Canny H.) 301
7.18. Entorno *visualizador interactivo,* objeto a detectar: la *calculadora*. 303
7.19. Entorno *visualizador interactivo,* objeto a detectar: la *caja amarilla*. . . . 304
7.20. Entorno *visualizador deteccion,* objeto a detectar: la *caja amarilla*.La combinación de las escenas umbralizadas de ambas etapas disminuye el número de falsos positivos de cada una de las dos etapas por separado. . . 306
7.21. Objetos utilizados para mostrar la capacidad de detección de los distintos canales. 307
7.22. Algunas de las escenas complejas utilizadas durante el experimento. . . . 310
7.23. Resultados obtenidos para el primer objeto. Se muestran los píxeles clasificados incorrectamente (falsos positivos, falsos negativos y suma de ambos) para todos los canales y para la media ponderada de los mismos. . 311
7.24. Resultados obtenidos para el segundo objeto. 312
7.25. Resultados obtenidos para el tercer objeto. 313
7.26. Resultados obtenidos para el cuarto objeto. 314
7.27. Resultados promediados para los cuatro objetos utilizados. Ninguno de los canales es capaz de ofrecer resultados óptimos sobre los cuatro objetos simultáneamente. 315
7.28. Resumen de los resultados obtenidos con la media ponderada de los mapas de probabilidad en la etapa multicanal PCA. 316
7.29. Resultados obtenidos con el canal LoG_{ON-OFF} utilizado en la etapa multiescala NMFSC. 318
7.30. Resultados finales tras combinar las etapas multicanal y multiescala. . . . 318

C.1. Modelos del objeto *la caja amarilla* en los canales de Apariencia H, S, V, I, r y g. 334
C.2. Modelos del objeto *la caja amarilla* en los canales de Apariencia b, l1, l2, l3, c1 y c2. 335
C.3. Modelos del objeto *la caja amarilla* en el canal de Apariencia c3 y en los canales de Histograma H, S, RGB, r y g. 336
C.4. Modelos del objeto *la caja amarilla* en los canales de Histograma b, l1, l2, l3, c1 y c2. 337
C.5. Modelos del objeto *la caja amarilla* en el canal de Histograma c3 y en los canales de Forma H, S, V, I y r. 338
C.6. Modelos del objeto *la caja amarilla* en los canales de Forma g, b, l1, l2, l3 y c1. 339
C.7. Modelos del objeto *la caja amarilla* en los canales de Forma c2 y c3. . . . 340
C.8. Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Apariencia H, S, V, I, r y g.. 342

C.9. Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Apariencia b, l1, l2, l3, c1 y c2. 343
C.10. Distribuciones espaciales de los modelos del objeto *la caja amarilla* en el canal de Apariencia c3 y en los canales de Histograma H, S, RGB, r y g. . 344
C.11. Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Histograma b, l1, l2, l3, c1 y c2. 345
C.12. Distribuciones espaciales de los modelos del objeto *la caja amarilla* en el canal de Histograma c3 y en los canales de Forma H, S, V, I y r. 346
C.13. Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Forma g, b, l1, l2, l3 y c1. 347
C.14. Distribuciones espaciales de los mModelos del objeto *la caja amarilla* en los canales de Forma c2 y c3. 348

Índice de tablas

2.1. J-invariantes teóricos del objeto considerado. 16
2.2. Resumen de los principales trabajos que resuelven el problema del reconocimiento visual de objetos basado en la apariencia utilizando PCA como método de extracción de características. 28
2.3. Principales trabajos que realizan comparativas del funcionamiento de PCA e ICA como método de extracción de características en tareas de reconocimiento visual de objetos basado en la apariencia. 29

3.1. Principales métodos de selección de características 46
3.2. Principales métodos de extracción de características 47
3.3. Clasificadores no supervisados: principales métodos de agrupamiento: agrupamientos basados en el criterio SSE y basados en grafo. 53
3.4. Conexiones entre los métodos estadísticos clásicos y los métodos basados en redes neuronales 54

4.1. Tasas de éxito en el reconocimiento para los objetos de la librería COIL-20 utilizando subespacio universal PCA en los canales: niveles de gris, bordes y binario 117
4.2. Tasas de éxito en el reconocimiento con oclusiones 123
4.3. Resultados experimentales de reconocimiento con distancia euclídea para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la COIL-20. 125
4.4. Resultados experimentales de reconocimiento con distancia de Mahalanobis para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la COIL-20. 125
4.5. Tasas de éxito en el reconocimiento con oclusiones para la arquitectura subespacio universal utilizando 30 vistas para el entrenamiento. 126
4.6. Resultados experimentales de reconocimiento con la arquitectura subespacio objeto modificado PCA para las imágenes de la COIL-20 en el canal I(niveles de gris) con el entrenamiento tipo *completo* 129
4.7. Resultados experimentales de reconocimiento con subespacios objeto PCA para las imagenes de la COIL-20 en el canal I(niveles de gris) con el entrenamiento tipo *aleatorio* 132

4.8. Resultados experimentales de reconocimiento con la arquitectura subespacio objeto modificado PCA para las imagenes de la COIL-20 en el canal Binario y con el entrenamiento tipo *completo*. 132
4.9. Resultados experimentales de reconocimiento con la arquitectura subespacio objeto modificado PCA para las imagenes de la COIL-20 en el canal Bordes y con el entrenamiento tipo *completo*. 132
4.10. Tasas de éxito en el reconocimiento para los objetos de la librería UMH-15 utilizando subespacio universal PCA en los canales: niveles de gris, bordes y binario. 136
4.11. Tasas de éxito en el reconocimiento para los objetos simétricos de la librería UMH-15 utilizando subespacio universal PCA en los canales: niveles de gris, bordes y binario. 137
4.12. Resultados experimentales de reconocimiento con distancia euclídea para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la UMH-15. 138
4.13. Resultados experimentales de reconocimiento con distancia de Mahalanobis para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la UMH-15. 138
4.14. Resultados experimentales de reconocimiento con distancia de Mahalanobis para la arquitectura subespacio objeto modificado PCA con las imágenes de la UMH-15. 139
4.15. Resultados experimentales de reconocimiento con distancia euclídea para la arquitectura subespacio objeto modificado PCA con las imágenes de la UMH-15. 140

5.1. Número de subconjuntos a testear en la selección exhaustiva de características. 157

A.1. Notación utilizada en el capítulo 2. 325
A.2. Notación utilizada en el capítulo 3. 326
A.3. Notación utilizada en el capítulo 4. 327
A.4. Notación utilizada en el capítulo 5. 327
A.5. Notación utilizada en el capítulo 6. 328
A.6. Notación utilizada en el capítulo 7 relativa a los canales utilizados. 329

Capítulo 1

Introducción

La presente tesis se centra en el análisis de las principales metodologías de reconocimiento de objetos basadas en la apariencia visual que existen en la actualidad. Dicho análisis ha permitido el desarrollo de un sistema de detección de objetos en entornos complejos fundamentado en los métodos de extracción de características que se han revelado como los más idóneos para dicha tarea.

1.1. Motivación y objetivos

Los motivos que han llevado a la realización de la presente tesis se pueden resumir en cuatro puntos fundamentales:

- ***El reconocimiento de objetos tridimensionales en entornos no estructurados mediante visión artificial es un problema aún por resolver***. Actualmente la visión por computador ha logrado avances considerables en el campo del reconocimiento de objetos visuales a partir de imágenes bidimensionales de los objetos que se pretenden identificar, cuando los objetos de interés se conocen muy bien a priori y el entorno del objeto está altamente controlado. El entorno suele ser el conjunto del fondo visual del objeto y la iluminación. Es posible denominar este tipo de reconocimiento visual, como reconocimiento de objetos en entornos *estructurados*. Ejemplos de aplicaciones de visión estructurada serían: sistemas de inspección de defectos sobre piezas manufacturadas, sistemas de medición de alta precisión, sistemas de reconocimiento automático de matrículas de coches, sistemas de supervisión de etiquetado durante el envasado de un producto, sistemas clasificadores de frutas para su almacenaje, sistemas guiadores de robots para la manipulación de piezas, sistemas de clasificación e inspección de botellas recicladas, etc. Muchas aplicaciones de este tipo están ya disponibles en la vida cotidiana como sistemas comerciales [1] [4]. Todas estas aplicaciones tienen una arquitectura de sistema de visión similar, el objeto está claramente definido, así como el entorno visual de trabajo y suelen implantarse en entornos industriales. Cuando estas circunstancias no se producen,

los sistemas automáticos no son capaces de funcionar adecuadamente. Curiosamente, los sistemas automáticos fallan precisamente en las tareas que con más sencillez realizan los humanos, cualquier persona puede reconocer con facilidad objetos (en sentido genérico, caras, coches, pequeñas piezas, etc.) con independencia del tipo de iluminación, del fondo de las escenas, de la distancia o posición relativa de tales objetos, etc. Un reto de la presente tesis es avanzar en el desarrollo de sistemas de reconocimiento de objetos cuando las circunstancias son similares a las indicadas: fondos no estructurados y condiciones desconocidas a priori.

Figura 1.1: Detalle del robot móvil B21r en un laboratorio de la universidad. Si se desea que el robot interactúe con los humanos ha de ser capaz de detectar caras y reconocerlas bajo distintos tipos de iluminación, fondos complejos y diferentes distancias respecto de su posición.

- ***El funcionamiento de múltiples aplicaciones está limitado en la actualidad por no existir sistemas fiables de reconocimiento de objetos.*** La falta de sistemas automáticos de reconocimiento de objetos suficientemente fiables limita actualmente muchas aplicaciones. En primer lugar, la robótica de servicio, que ha experimentado grandes avances en las últimas décadas, sigue encontrando una barrera infranqueable en las capacidades visuales de los robots. Para lograr un comportamiento autónomo en un entorno no estructurado, un robot debería ser capaz de indentificar los objetos que encuentran a su alrededor. Mientras esto no sea posible, las aplicaciones de los robots estarán altamente limitadas. Además de la robótica, otras

muchas aplicaciones están también limitadas por la falta de técnicas fiables de reconocimiento de objetos, como por ejemplo los sistemas de vigilancia autónomos.

- ***Controversia entre métodos basados en modelo y métodos basados en apariencia.*** Un estudio de los trabajos realizados en el campo del reconocimiento de objetos muestra dos tendencias completamente distintas: el reconocimiento basado en modelo y el reconocimiento basado en apariencia. Ambas técnicas tienen sus ventajas y sus inconvenientes. Aparentemente, el reconocimiento basado en modelo podría ser potencialmente más eficaz y permitiría identificar objetos en cualquier posición; pero plantea importantes problemas de aplicabilidad práctica. En la presente tesis se discuten ambas opciones y se concluye que el reconocimiento basado en la apariencia puede ofrecer mejores resultados prácticos, por lo que es el método que se estudia con profundidad y sobre el que se centran todos los trabajos realizados.

- ***La literatura existente presenta grandes contradicciones en cuanto a las técnicas de extracción de características más adecuadas.*** El reconocimiento basado en la apariencia se apoya en técnicas de extracción de características o compresión de la información. La literatura muestra muchas contradicciones en cuanto a las técnicas más adecuadas, sin presentar justificaciones teóricas claras sobre las ventajas de cada método. En la presente tesis se intenta aportar algo de luz en este aspecto, particularmente en lo relativo a la aparente superioridad del análisis de componentes independientes sobre el análisis de componentes principales en la extracción de características para el reconocimiento visual de objetos.

De acuerdo con esta serie de motivaciones, los objetivos que se pretenden alcanzar con la presente tesis son los siguientes:

- Aclaración de la controversia entre ICA y PCA como métodos de extracción de características en el reconocimiento basado en la apariencia y propuesta de métodos alternativos.

- Conseguir un sistema de detección de objetos fiable ante cambios de iluminación, fondos o pose de los objetos a detectar.

- Utilización de la apariencia global de las escenas y del aprendizaje a partir de ejemplos para la detección de los objetos.

- Conseguir total autonomía o mecanismo automáticos de ajuste de los parámetros que afecten a la detección.

- Reducción de la velocidad de cómputo a valores adecuados para una implementación en tiempo real.

1.2. Marco de la tesis

La presente tesis se ha desarrollado bajo el marco de dos proyectos de investigación, ambos centrados en el campo de la robótica y la visión por computador.

El primer proyecto lleva por título *Automatización de tareas complejas en robótica móvil mediante sistemas inteligentes RAMI (Robot Autónomo Móvil Inteligente)*; y fue financiado a través de la Conselleria de Innovación y Competitividad de la Generalitat Valenciana con referencia CTDIA/2002/03. El segundo proyecto, *Herramientas de Teleoperación Colaborativa. Aplicación al Control Cooperativo de Robots*, vigente hasta el 2007, está financiado por la Comisión Interministerial de Ciencia y Tecnología con referencia DPI2004-07433-C02-01. En ambos proyectos se proponen, entre otras, herramientas de localización y de navegación basadas en la cooperación entre robots, para lo cual, el reconocimiento de objetos es tarea imprescindible.

La mayor parte de los desarrollos se han llevado a cabo en las instalaciones de investigación del grupo de *Automatización, Robótica y Visión por Computador*, perteneciente al Departamento de Ingeniería de Sistemas Industriales de la Universidad Miguel Hernández. El otro centro en el que se han realizado trabajos ha sido el Departamento de Informática de la Universidad de Helsinki, concretamente en el grupo de investigación *Neuroinformatics* liderado por el Dr. Aapo Hyvärinen. En este centro, en el curso de una estancia de tres meses realizada entre Agosto y Noviembre de 2004, se llevaron a cabo los estudios y experimentos relativos al uso de ICA como método de extracción de características para los sistemas de reconocimiento de objetos basado en la apariencia. Específicamente, el trabajo se centró en el análisis de dos de los algoritmos de ICA más populares en la actualidad: *FastICA* e *Infomax*. Se analizaron tanto teórica como experimentalmente las supuestas ventajas que aportan los vectores de componentes independientes frente a los vectores de componentes principales para la clasificación de objetos. Los resultados que se muestran en la sección 5.1 son fruto de tal estudio.

1.3. Resumen de las aportaciones realizadas

En los últimos años han aparecido numerosos trabajos en los que se afirma la superioridad de la técnica de ICA (*Independent Component Analysis*) frente al método tradicional de PCA (*Principal Componente Analysis*), aunque también existen fuentes que afirman la idea opuesta. Una de las principales aportaciones de esta tesis es la realización de un estudio analítico y experimental de las similitudes y diferencias de ambos métodos y la justificación de las circunstancias en las cuales ICA puede superar los resultados de PCA. Parte de los resultados obtenidos en dicho estudio se concentran en las publicaciones [160], [169], [171], [170], [172].

Otra de las aportaciones de la presente tesis es la propuesta de aplicabilidad del método RP (*Random Projection*) como método extractor de características para reconocimiento de objetos. Aunque sus tasas de éxito en el reconocimiento son inferiores a las de otros métodos, presenta otras ventajas como la simplicidad de su generación *off-line* independiente de la naturaleza de los datos ([166], [167]).

En este documento también se describen los resultados de un método muy reciente de extracción de características, NMFSC (*Non-negative Matrix Factorization with Sparseness Constraints*), bajo una nueva arquitectura de reconocimiento de objetos, arquitectura subespacio objeto modificado, siendo su principal ventaja la capacidad para producir representaciones locales de los objetos, lo que permite mejorar el reconocimiento de los objetos en entornos no estructurados, es decir, en presencia de una gran cantidad de objetos, oclusiones o iluminación variable ([161], [162]).

También se propone de forma totalmente original un método de detección de objetos basado en apariencia multicanal y multiescala, cuya característica principal es su capacidad para trabajar en entornos no estructurados, es decir, sin ningún tipo de control en la geometría o apariencia del objeto, la posición del mismo respecto de la cámara, el fondo de la escena, la iluminación del entorno y la presencia de oclusiones o agrupaciones con otros objetos [165].

El método de detección propuesto presenta los siguientes aspectos novedosos: realiza la combinación de dos etapas de búsqueda en paralelo; hace uso de dos métodos de extracción de características distintos: PCA y NMFSC; utiliza múltiples canales de información y pondera su peso en la detección final de forma automática; realiza la combinación de los resultados en función del objeto a detectar gracias a un proceso de entrenamiento automático; y utiliza escalas o resoluciones distintas para facilitar la búsqueda en entornos complejos donde se desconoce la distancia a la que se encuentra el objeto de interés.

1.4. Estructura de la tesis

La tesis se estructura de la forma siguiente:

El capítulo dos se dedica al análisis de las dos tendencias fundamentales, modelo y apariencia, que existen en el ámbito actual del reconocimiento de objetos. La conclusión del tal estudio es que los métodos basados en apariencia son más adecuados para una utilización en entornos reales. En base a esta conclusión, se analizan las ventajas e inconvenientes de los principales trabajos previos que realizan reconocimiento basado en apariencia.

En el capítulo tres se presenta el esquema básico de un sistema de reconocimiento de patrones para luego seguir con la descripción detallada de algunas de las técnicas de extracción de características más utilizadas y cuyo funcionamiento se pretende analizar en el marco de los sistemas de reconocimiento de objetos basados en la apariencia.

En el capítulo cuatro se propone una nueva arquitectura de reconocimiento de objetos basada en la arquitectura *objeto* original de Murase [118]. La nueva propuesta permite no sólo detectar orientaciones sino también identificar objetos. Además de detallar su metodología, tambien se presentan los resultados experimentales obtenidos con las arquitecturas clásicas y la propuesta en esta tesis bajo el marco de la técnica del análisis de componentes principales.

En el capítulo cinco se estudian analítica y experimentalmente los métodos de extracción de características expuestos en el capítulo tres en el ámbito del reconocimiento visual de objetos. Básicamente se analizan los métodos de análisis de componentes independientes, *random projection* y *non-negative matrix factorization* frente al método clásico de análisis de componentes principales, justificando los resultados aparentemente contradictorios de muchos de los estudios comparativos previos.

En el capítulo seis se describe el sistema de detección de objetos basado en apariencia multicanal y multiescala propuesto en la presente tesis, cuya característica principal es su capacidad para trabajar en entornos no estructurados. El funcionamiento del sistema de detección de objetos propuesto se detalla en el capítulo siete.

Finalmente, en el apartado de conclusiones, se discute la validez de la propuesta presentada en la tesis, así como las aportaciones presentadas y su aplicabilidad práctica. Por último, se plantean los posibles trabajos futuros que pueden realizarse sobre el marco de detección de objetos propuesto.

Capítulo 2

Reconocimiento de objetos mediante visión por computador

La presente tesis se centra en el estudio de los sistemas de reconocimiento basados en su apariencia visual. En la actualidad, estas técnicas de reconocimiento proporcionan mejores resultados que las técnicas basadas en modelo en cuanto a tasas de éxito en el reconocimiento, capacidad de reconocimiento de objetos de forma libre, manejo de cambios en la iluminación, oclusiones y uso de escenarios con fondos no estructurados. Sin embargo, en general, sus principales desventajas son su alto coste computacional y que hacen uso de gran cantidad de memoria para almacenar los modelos de los objetos a reconocer.

En este capítulo se analizan los factores anteriores mediante un estudio en profundidad de las diferentes alternativas existentes para el reconocimiento de objetos, tanto basadas en modelo como basadas en apariencia. La conclusión del tal estudio es que los métodos basados en apariencia son más adecuados para una utilización en entornos reales. En base a esta conclusión, se analizan las ventajas e inconvenientes de los principales trabajos previos que realizan reconocimiento basado en apariencia.

2.1. Teorías Computacionales de Reconocimiento de Objetos

El problema general de *reconocimiento de objetos* es demasiado amplio para su descripción global. De este modo, en la literatura actual, el concepto de *reconocimiento de objetos* es posible plantearlo bajo tres formas o subproblemas según el tipo de restricciones que sean impuestas:

- **Reconocimiento de Objetos Específicos**: En un escenario típico, una imagen de test muestra exactamente un objeto de interés (con o sin oclusiones y/o con o sin agrupamientos con otros objetos). La tarea es reconocer ese objeto como uno de los posibles objetos a reconocer [13][118], por ejemplo un juguete o una landmark particular. La apariencia del objeto en la librería o base de datos es conocida con precisión.

- **Reconocimiento o Clasificación de Objetos Genéricos**: El escenario es el mismo que en el caso precedente, pero ahora la tarea es etiquetar un objeto como perteneciente a una de las clases o categorías [179]. Los objetos dentro de una clase pueden variar ligeramente en apariencia. Una clase determinada (*coche*, *cara*, *silla*) puede ser definida de varias formas: mediante prototipos, descripciones o modelos que contengan sufiente información y detalle para discriminar entre las diferentes clases pero no diferenciar entre objetos dentro de una clase.

- **Detección de objetos**: Las imágenes a testear contienen diversos objetos y en general, muestran una escena real. La tarea es localizar regiones en la imagen que muestren al objeto de interés (*i.e.* un detector de caras humanas)[173].

La tarea de detección de objetos engloba los dos otros casos, puesto que el prentender detectar un objeto en una escena compleja conlleva implícito la tarea del reconocimiento de objetos específicos;tarea que se convierte en el segundo caso en el momento que las condiciones del entorno son variables, como la iluminación de la escena o los parámetros de la cámara utilizada para la adquisición de la imagen.

En esta sección se revisan los principales métodos visuales de reconocimiento de objetos que existen en la actualidad, realizando una clasificación de los mismos según la teoría computacional utilizada para la representación del objeto [108]. De esta forma, se suele distinguir entre *métodos estructurales* o *basados en modelo* y métodos *basados en la apariencia* [130], aunque clasificaciones más detalladas como la proporcionada por Edelman en [49] subdivide la clasificación anterior en más tipos, así, dentro de los métodos estructurales es posible distinguir entre *descripciones estructurales* y *restricciones geométricas*.

Los métodos estructurales realizan la discriminación de objetos utilizando las relaciones geométricas presentes en los objetos. La idea fundamental es la utilización de estructuras de datos simbólicas tales como cadenas, árboles, grafos,.. en lugar de vectores de números que se utilizan en otros métodos. Las estructuras de datos simbólicas facilitan los modelos jerárquicos que describen cómo un patrón o subpatrón es construido a partir de partes más simples. El reconocimiento de un patrón desconocido se lleva a cabo comparando su representación simbólica con un número de modelos de objeto predefinidos. Esos modelos pueden proporcionarse en términos de prototipos estructurales o gramáticas y el proceso de comparación se basa en la correspondencia simbólica o sintáctica.

Por otra parte, los métodos basados en la apariencia utilizan toda la información presente en las imágenes de los objetos en lugar de propiedades o características. Por consiguiente, se dispone de una base de datos donde tendremos almacenadas las imágenes de los objetos. Para el reconocimiento de un objeto concreto se dispone de un conjunto de imágenes sobre las posibles apariencias de dicho objeto; esto es, el conjunto de imágenes tomadas, idealmente, a partir de todos los posibles puntos de vista y con todas las variaciones posibles de iluminación.

Para la mejor comprensión de la variedad de problemas que surgen en conexión con los distintos métodos de reconocimiento de objetos, cabe realizar dos distinciones básicas: entre *percepción* y *reconocimiento*; y entre *reconocimiento* y *clasificación*. Es necesario distinguir entre la *percepción* de la forma (o apariencia) de un objeto observado y el *reconocimiento* de objetos a partir de sus formas (o apariencias).

Esta observación fue realizada por primera vez por Wittgenstein [175] que intentó resaltar la diferencia entre ver una forma y verla como algo. A diferencia de simplemente percibir una forma, reconocerla implica memoria, es decir, representaciones de formas vistas en el pasado. La forma de esas representaciones viene condicionada por diversos factores como la iluminación y la orientación que afectan a la apariencia de los objetos. Debido a los efectos de la orientación, por ejemplo, simplemente almacenando una vista particular del objeto para futuras referencias, el reconocedor no funcionaría del todo bien, pues otra vista del mismo objeto puede ser menos similar a la vista del objeto almacenada y más semejante a otro objeto diferente, dando lugar a un reconocimiento erróneo. Los métodos actuales de reconocimiento, sobre todo los basados en apariencia [99] [118] [124] , consideran este problema y son capaces de superarlo, si es posible disponer de varias vistas del mismo objeto y son utilizados clasificadores adecuados.

Estos algoritmos, sin embargo, no consideran necesariamente, la otra distinción entre *reconocimiento* y *clasificación*. En la inmensa mayoría de los métodos de reconocimiento, el *reconocimiento* se asocia a recordar algo visto anteriormente, mientras que la clasificación implica asignar una clase a un objeto que nunca se ha visto. Para ser capaz de reconocer un objeto nuevo, esto es clasificar, es necesario que el sistema de reconoci-

miento considere, además de memoria, procesos para abstraer nuevos objetos a partir de los almacenados. Esta capacidad se ha desarrollado y solucionado, al menos de forma teórica, en algunos algoritmos estructurales como los presentados en [19] y [109]; sin embargo, no ha sido del todo cubierta por los métodos basados en la apariencia.

En términos generales, es posible afirmar que los algoritmos basados en modelo actuales son teóricamente capaces de realizar la tarea de clasificar nuevos objetos, desconocidos por el sistema, pero en la práctica no permiten su utilización en sistemas visuales de reconocimiento reales debido a la imposibilidad de segmentar de la imagen los datos simbólicos necesarios para su funcionamiento y por lo tanto son incapaces de funcionar en un entorno real. Por otro lado, los algoritmos de reconocimiento basados en la apariencia, sí son realizables en sistemas reales, siempre que la base de modelos de objetos se haya realizado adecuadamente, por lo que separan correctamente las aptitudes de percepción y reconocimiento. Sin embargo, la tarea de clasificación, aún no es conseguida con la suficiente tasa de éxito. En [116] se presentan los resultados experimentales de comparar un método estructural con uno basado en la apariencia, obteniéndose las conclusiones antes expuestas: el algoritmo basado en la apariencia es incapaz de reconocer objetos no vistos anteriormente y el sistema estructural sólo es capaz de segmentar las primitivas necesarias para el reconocimiento bajo entornos estructurados. Si el objeto se encuentra sobre un fondo complejo, la extracción de primitivas adecuadas es muy compleja o da lugar a un conjunto muy amplio de éstas.

A continuación, se detallan algunos de los más importantes métodos de reconocimiento presentando sus ventajas e inconvenientes, según la clasificación anterior.

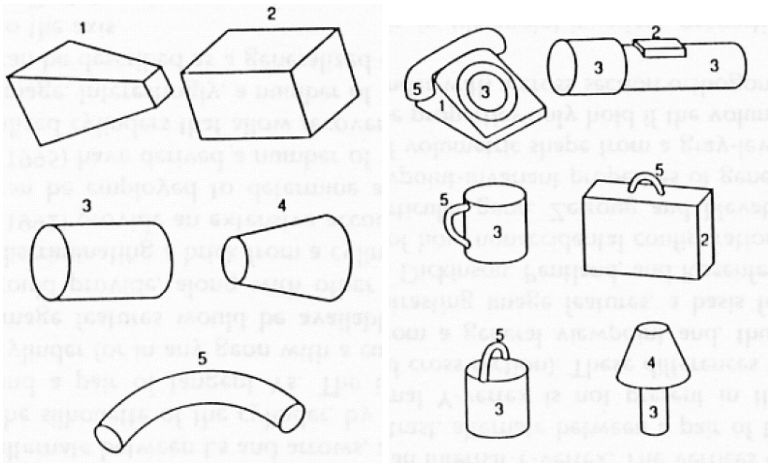
2.1.1. Métodos basados en modelo

2.1.1.1. Descomposiciones estructurales

En cualquier método basado en modelo de descomposición estructural, el objeto se describe en términos de unos cuantos componentes básicos unidos entre sí mediante relaciones espaciales seleccionadas de un conjunto fijo. La generalización de las primitivas (los componentes y sus relaciones) es crucial para el éxito en la representación de nuevos objetos. Matemáticamente, la comparación entre objetos equivale a una correspondencia de etiquetas; y la clasificación, a la atribución de un objeto a una clase de etiquetas equivalentes a objetos estructuralmente idénticos, aunque geométricamente distintos.

La teoría de descomposición estructural más conocida es la teoría RBC[1] o *reconocimiento por componentes* de Biederman [19], la cual postula un conjunto de unos treinta componentes básicos denominados *geones* y que teóricamente son fácilmente detectables en las imágenes debido a sus propiedades *no accidentales*, esto es, características 3D que

[1] Recognition By Components (RBC).



(a) (b)

Figura 2.1: En la figura de izqda. se muestra un conjunto de 5 *geones* y en la dcha., unos objetos representados según la teoría RBC.

son casi siempre preservadas durante el proceso de la proyección del objeto en la imagen [106]. El uso de características no accidentales para inferir la presencia de geones, y el cálculo de la estructura del objeto descompuesto en geones son las bases del método RBC descrito en [19]. La figura 2.1 muestra un conjunto reducido de geones y unos cuantos objetos descritos según la teoría RBC. En su trabajo, Biederman demuestra la capacidad de una red neuronal multicapa cuidadosamente diseñada para derivar representaciones estructurales a partir de geones compuestos por trazos lineales. Sin embargo, su método tiene una serie de fuertes inconvenientes que impiden el uso de la teoría RBC en la práctica y que a continuación se enumeran:

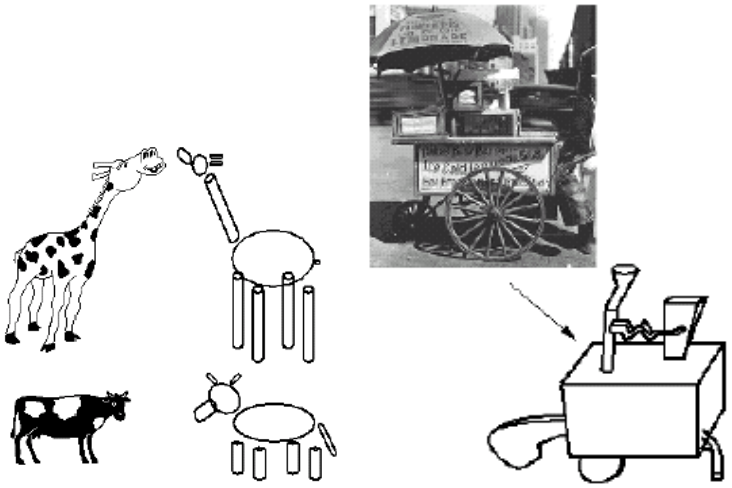
- *Carencia de información métrica*: Aunque el método RBC es en principio capaz de representar nuevos objetos (a través de su descomposición estructural), ignora las diferencias métricas entre geones equivalentes. Este defecto fue señalado y enmendado por Stankiewick y Hummel [151], quienes expandieron la teoría RBC con variables cuantitativas, codificando las longitudes de las diversas partes de un objeto. La figura 2.2(a) ilustra este problema.

- *Imposibilidad de segmentación de los geones*: El problema más severo al que se enfrenta la teoría RBC es la necesidad de detectar fielmente los componentes en las imágenes.Una parte del problema de esta detección (inferir la estructura 3D a partir de proyecciones 2D) está resuelto mediante el uso de características no accidentales. Sin embargo, la dificultad de encontrar en la imagen líneas y uniones cuyas relaciones no accidentales se puedan utilizar para inferir la presencia de geones ha excluido a la teoría RBC de ser aplicada en la práctica para el reconocimiento de

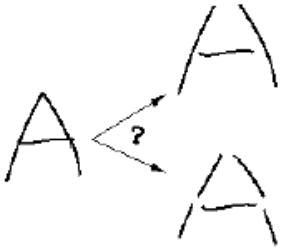
objetos en imágenes. El modelo descrito en [19] funcionaba con líneas etiquetadas manualmente. Un ejemplo gráfico de este inconveniente de los métodos estructurales se muestra en la figura 2.2(b).

- *Inestabilidad de la descripción estructural*: Incluso en el caso en que la entrada al sistema de descomposición estructural sea una colección de líneas etiquetadas, su interpretación es, en la mayoría de los casos, problemática debido a una inestabilidad inherente del enfoque estructural. Esta inestabilidad surge de la posibilidad de descomponer cualquier objeto de varias maneras diferentes según las primitivas que se asuman. Por ejemplo la letra `A` de la figura 2.2(c) puede ser descompuesta en 3 o 5 líneas rectas según si los lados de `A` se representen como líneas sencillas o concatenaciones de 2 segmentos más cortos. Es fácil darse cuenta que este mismo problema surge en cualquier dominio estructural.

  Para resolver el problema de la inestabilidad de la descripción estructural se han propuesto soluciones en las que se impone una probabilidad a priori (en el sentido bayesiano) para cada posible solución [20][64].



(a) (b)



(c)

Figura 2.2: Inconvenientes de las representaciones estructurales. Fig. 2.2(a): la inclusión de detalles métricos reduce la habilidad para reconocer nuevos objetos. Fig. 2.2(b): Una imagen de un carrito de *hot-dogs* y su objecto en representación RBC según Biederman [19]. Actualmente es imposible segmentar en la imagen original de niveles de gris las primitivas que generan los geones. Fig. 2.2(c): Incluso en las tareas de reconocimiento más simples como el reconocimiento de caracteres, derivar una descripción estructural única es problemático.

2.1.1.2. Restricciones geométricas

Los métodos basados en restricciones geométricas se fundamentan en la restricción de consistencia del punto de vista, esto es, el establecimiento de correspondencias entre características localizadas del objeto y las restricciones en la imagen de los lugares donde residen las características, es decir, se basan en la geometría del objeto.

Dentro de estos métodos, aquí destacamos aquellos algoritmos de reconocimiento de objetos basados en invariantes proyectivos [53][71]. Los parámetros que definen una imagen pueden variar debido a las diferentes condiciones de luz, de calibración de la cámara y de la posición de ésta. Es por tanto deseable buscar propiedades geométricas del objeto que permanezcan invariantes bajo tales cambios. Si el proceso de captación de imágenes por una cámara es modelado como una transformación proyectiva, los invariantes proyectivos pueden servir como esos parámetros que permitan identificar el objeto a reconocer en la imagen. Los invariantes proyectivos pueden clasificarse, en general, en 3 grupos básicos, según si los puntos y/o rectas características de los objetos residen o no en un solo plano [2]:

Invariantes coplanares:

- Cross-Ratio o Razón Doble para 4 puntos colineales
- Invariante de 5 puntos coplanares y J -Invariante de 5 rectas coplanares
- Invariante de 2 rectas coplanares y de 2 puntos
- Invariante de Dos Cónicas Coplanares
- Invariante de Cónica y puntos:razón doble de cuatro puntos
- Invariante de Cónica y rectas:razón doble de cuatro rectas

Invariantes de varios planos

- Invariante de Dos Cónicas No Coplanares

Invariantes espaciales

- Invariante espacial
- Invariante de P^n

Por ejemplo, sería posible reconocer el objeto de la figura 2.3 mediante el cálculo del J-invariante de 5 puntos coplanares [159]. La cara interesante del objeto es la formada por las esquinas GHIJKL, al disponer de 6 puntos-esquina. El resto de las caras sólo tiene 4, por lo que no sirven para el cálculo de este invariante de 5 puntos.

[2]Para una descripción más detallada de los invariantes que aqui se enuncian, se recomienda consultar [159] y [168].

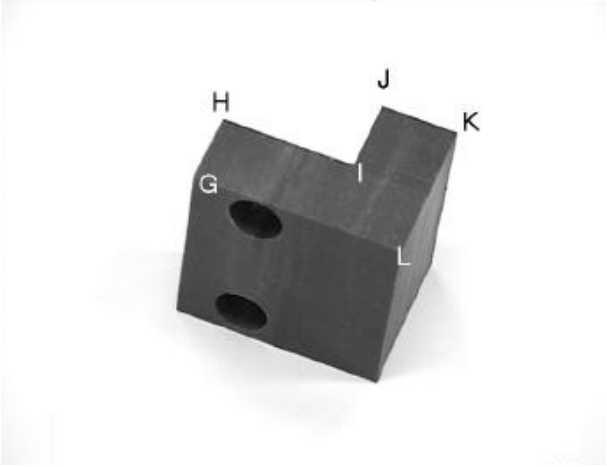


Figura 2.3: Objeto a reconocer: esquinas numeradas con letras. La cara más significativa del objeto es la formada por las esquinas GHIJKL al disponer de 6 puntos-esquinas

Las medidas físicas de la pieza proporcionan las coordenadas reales de la cara útil de la pieza (situando el eje en la esquina superior izquierda de la pieza):

$$\begin{aligned} G &= [0, 58{,}8] \\ H &= [0, 30] \\ I &= [49{,}6, 30] \\ J &= [49{,}6, 0] \\ K &= [78{,}85, 0] \\ L &= [78{,}85, 58{,}8] \end{aligned} \tag{2.1}$$

En la siguiente tabla aparecen los J-invariantes teóricos que se pueden utilizar como modelos geométricos del objeto, obtenidos a partir de las medidas físicas de la pieza, de las 6 combinaciones posibles de 6 puntos de la cara útil del objeto:

J-invariante (G,H,I,J,K) = 2.1865 2.2977 2.5687 2.7289 2.7982
J-invariante (H,I,J,K,L) = 2.1204 2.2057 2.5687 2.5846 2.7982
J-invariante (G,I,J,K,L) = 2.1865 2.2977 2.5687 2.7289 2.7982
J-invariante (G,H,J,K,L) = 2.1204 2.2057 2.5687 2.5846 2.7982
J-invariante (G,H,I,K,L) = 2.1865 2.2977 2.5687 2.7289 2.7982
J-invariante (G,H,I,J,L) = 2.1204 2.2057 2.5687 2.5846 2.7982

Tabla 2.1: J-invariantes teóricos del objeto considerado. .

Sería posible reconocer dicho objeto a partir de otras imágenes, si se obtienen los puntos-esquina necesarios para el cálculo de los invariantes. La figura 2.5 muestra algunas imágenes del objeto en poses diferentes con las esquinas significativa señaladas.

La figura 2.4 muestra los valores de los J-invariantes (G,H,I,J,K) obtenidos partir de los valores de las esquinas señaladas en las imágenes de la figura 2.5. Es posible apreciar como los valores obtenidos son muy similares al teórico que se utiliza como modelo de reconocimiento. La principal desventaja de este método de reconocimiento de objetos radica en la dificultad de obtener, a partir de la imagen, las esquinas necesarias para el cálculo del invariante proyectivo. Puesto que la existencia de sombras, cambios en la iluminación u oclusiones dificultan considerablemente la correcta extracción de las esquinas de interés.

Objeto		***J-invariante teórico (G,H,I,J,K)*** 2.1865 2.2977 2.5687 2.7289 2.7982				
posición	***combinación de puntos***	***J-invariante***				
vista1 izquierda	G,H,I,J,K	2.2746	2.4367	2.5170	2.7906	2.7999
vista1 derecha	G,H,I,J,K	2.2350	2.3074	2.6240	2.7652	2.7727
vista2 izquierda	G,H,I,J,K	2.2369	2.3104	2.6225	2.7674	2.7725
vista 2 derecha	G,H,I,J,K	2.2151	2.3138	2.5894	2.7575	2.7894
vista 3 izquierda	G,H,I,J,K	2.1803	2.3585	2.4802	2.7586	2.7957
vista 3 derecha	G,H,I,J,K	2.1873	2.3529	2.4980	2.7602	2.7988
vista 4 izquierda	G,H,I,J,K	2.1954	2.3158	2.5579	2.7463	2.7982
vista 4 derecha	G,H,I,J,K	2.2006	2.3280	2.5498	2.7561	2.7984

Figura 2.4: Invariantes obtenidos a partir de las esquinas señaladas de la figura 2.5 .

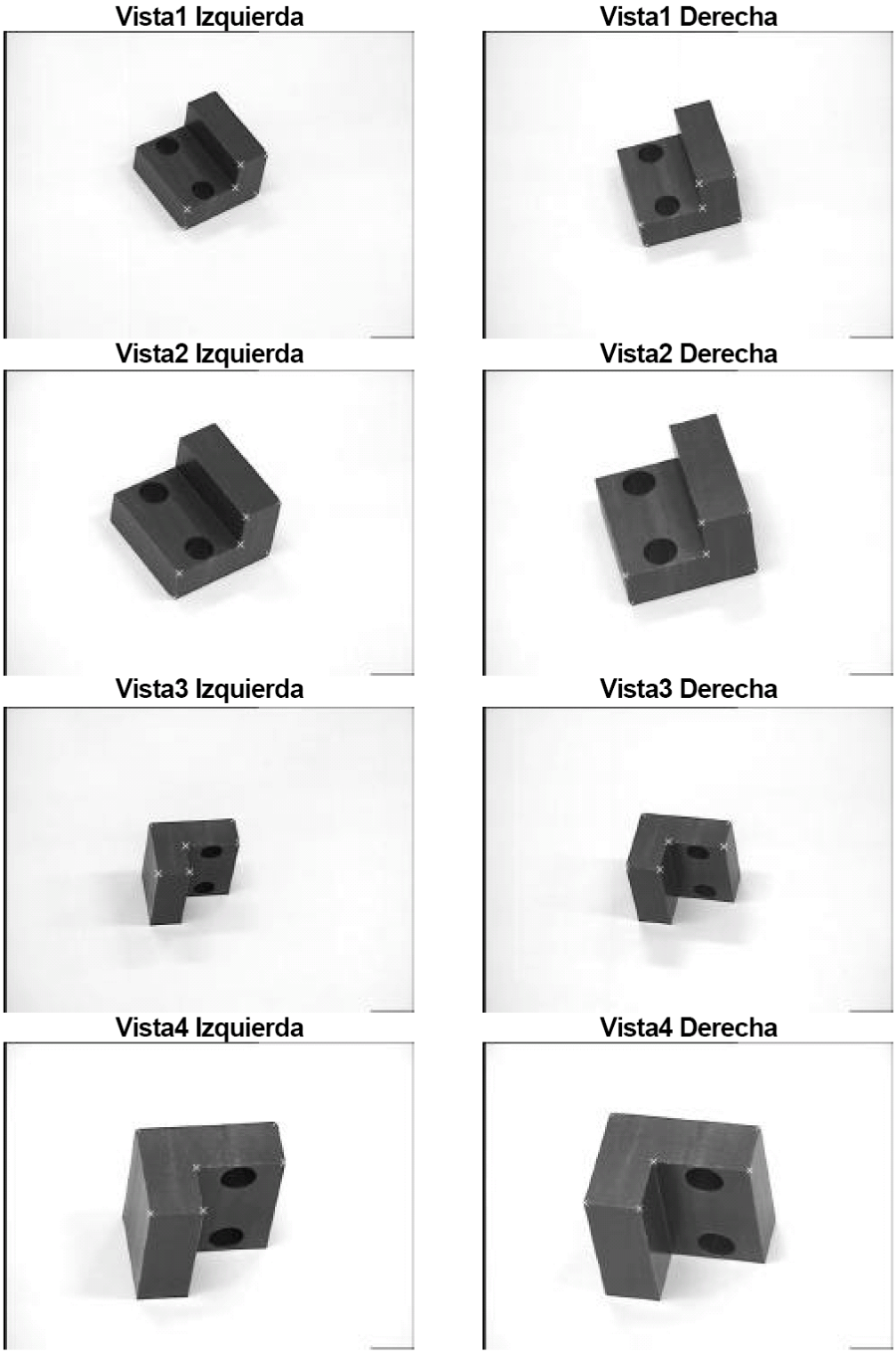
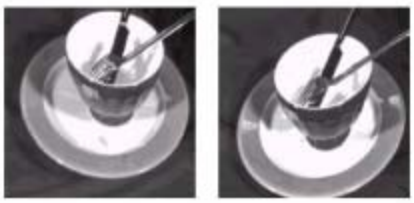


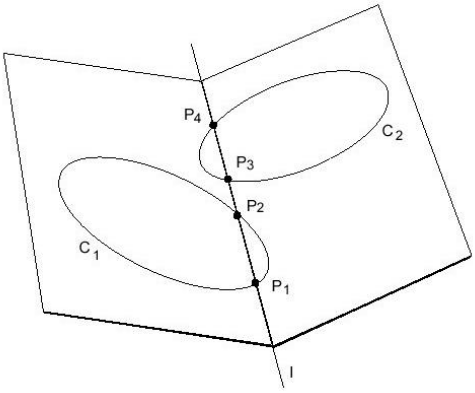
Figura 2.5: Diferentes vistas del objeto con las esquinas significativas señaladas.

Otro ejemplo de sistema de reconocimiento de objetos basado en invariantes geométricos se describe en [107], donde se utiliza el invariante de dos cónicas no coplanares para identificar dos cónicas en una imagen, partiendo de una escena real. En el experimento, se calcula primero la matriz fundamental a partir de las matrices de proyección de las cámaras calibradas anteriormente. Las cónicas son reconstruidas proyectivamente a partir de la matriz fundamental y por último se calculan los invariantes. La detección de cónicas (y de los bordes de la imagen en general) se realiza mediante el detector de bordes de Canny [31]. La figura 2.6 ilustra este tipo de reconocimiento de objetos basado en invariantes proyectivos.

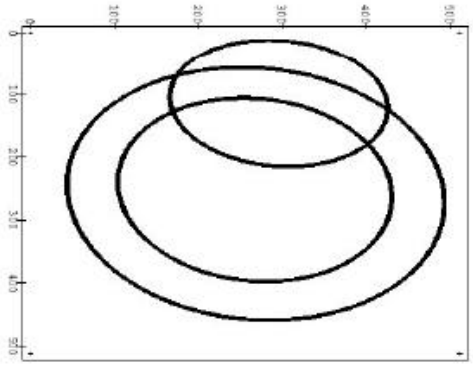
Como se ha descrito anteriormente, una de las mayores desventajas de los métodos basados en invariantes proyectivos es la dificultad de extraer las primitivas necesarias para la construcción de los invariantes, es decir, la detección de los puntos y rectas de los objetos que componen su geometría y permiten su reconocimiento. Además, el uso de invariantes proyectivos limita el tipo de objetos reconocibles, puesto que sólo aquellos que dipongan de las restricciones geométricas impuestas por los invariantes serán modelables.



(a) Par estereo de imagenes.



(b) Representación del invariante de dos cónicas no coplanares.



(c) Ajuste de las cónicas.

	bowl and dish outside	bowl and dish inside	dish inside and dish outside
1	-17.9643	13.6358	11.1248
2	2.1267	0.6532	4.42365
3	2.1136	0.676919	4.42377
4	**4.0192**	**4.01421**	**4.0008167**

(d) Resultados de reconocimiento.

Figura 2.6: Detalles del reconocimiento de objetos basado en el uso de invariantes de dos cónicas no coplanares[107]: a)imágenes estereoscópicas del objeto a reconocer, b)representación esquemática del invariante utilizado; c)gráfico con las cónicas detectadas; d)cálculo de los modelos que identifican al objeto, la cuarta fila es el valor del invariante proyectivo utilizado en distintas configuraciones del objeto.

2.1.2. Métodos basados en apariencia

Los sistemas de reconocimiento basados en la apariencia visual se han convertido en uno de los métodos más populares para el reconocimiento de objetos y la localización de sus poses en Visión por Computador. El éxito de estos métodos es debido en parte a su capacidad de reconocer sin dificultad objetos de forma libre y manejarse con los problemas de cambios de forma, pose, reflectancia y cambios en la iluminación de los objetos. En la siguiente sección se estudiará detalladamente en qué consisten los métodos basados en la apariencia.

2.2. Estructura de los sistemas de reconocimiento de objetos basados en la apariencia visual

Los sistemas de reconocimiento de objetos basados en apariencia suelen basarse en el concepto de *espacio imagen* o *espacio original*. Una imagen 2D, $I(x, y)$, puede representarse como un punto (o vector) de un espacio multidimensional, denominado espacio imagen, donde cada componente o dimensión espacial corresponde a un píxel de la imagen. Por ejemplo, una imagen de 64 filas por 64 columnas es equivalente a un punto de un espacio imagen de 4096 dimensiones. En general, una imagen de f filas y c columnas se considera un punto de un espacio imagen de M dimensiones, donde $M = f \times c$. Esta representación enmascara las relaciones de vecindad (distancias en el plano imagen) inherentes en una imagen 2D, es decir, el hecho de reordenar los píxeles de la imagen en un solo vector (y cambiar por lo tanto las relaciones de vecindad entre píxeles) no tiene efectos prácticos en su representación en el espacio imagen, puesto que todas las imágenes sufren la misma reordenación. Las operaciones espaciales como detección de bordes, filtrado y traslación no son operaciones locales en el espacio imagen. Por ejemplo, un filtro espacial 3×3 no es una operación tan eficaz en el espacio imagen, puesto que ahí es en realidad una multiplicación por una matriz enorme de $M \times M$ dimensiones compuesta en su mayoría por ceros. Por otro lado, la representación de los objetos en el espacio imagen ayuda a esclarecer las relaciones entre colecciones de imágenes de los objetos.

Al utilizar esta representación, las imágenes de los objetos se convierten en *vectores de características multidimensionales*, y por lo tanto es posible utilizar los métodos tradicionales de reconocimiento de patrones basados en características. Dicho de otro modo, considerando una imagen simplemente como un vector, los métodos basados en características pueden utilizarse para la tarea de reconocimiento visual basado en la apariencia, es decir, operaciones típicas de vectores de características tales como extracción, selección, clustering y medidas de distancias pueden aplicarse sobre las imágenes directamente. Sin embargo, la dimensionalidad tan elevada del espacio imagen produce que muchas operaciones de características sean inviables computacionalmente, puesto que conforme aumenta la resolución de una imagen, también aumenta la dimensión del es-

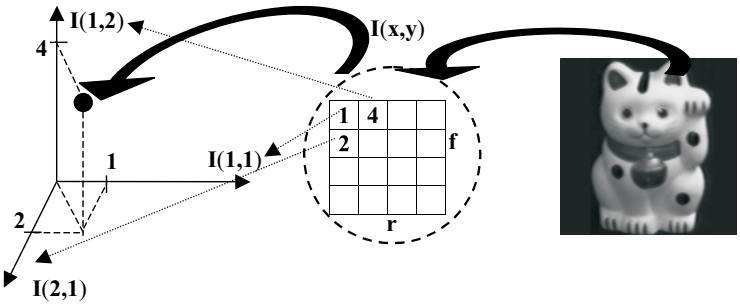


Figura 2.7: En el *espacio imagen* las imágenes de los objetos son puntos pertenecientes a un espacio multidimensional $M = f \times c$.

pacio imagen. En el límite, una imagen contínua puede mapearse en un espacio imagen de infinitas dimensiones. Afortunadamente, los cálculos básicos necesarios utilizan una dimensión igual al número de imágenes ejemplo y no al número de píxeles, permitiendo el reconocimiento incluso con imágenes de muy alta resolución

Si una imagen de un objeto es un punto del espacio imagen, una colección de N imágenes es equivalente a N puntos del espacio imagen, y pueden considerarse como muestras de una distribución de probabilidad. Es posible imaginarse todas las posibles imágenes o vistas de un objeto (bajo todo tipo de condiciones de iluminación, escala,..) y definir una *variedad* o *clase* para cada objeto dentro del espacio imagen (ver figura 2.8). El espacio imagen y la variedad de un objeto son representaciones matemáticas de muy alta dimensionalidad. Intuitivamente se puede realizar una estimación de la inmensidad del espacio imagen: si consideramos una imagen binaria de 8×8 dimensiones o píxeles (lo que es una imagen muy pequeña y con sólo dos valores posibles para cada píxel), el número de puntos en el espacio imagen (o el número de imágenes distintas) en este espacio imagen es 2^{64}. Si tenemos en cuenta que un ordenador actual de considerable potencia puede evaluar unos mil millones de imágenes por segundo, le llevaría unos 600 años evaluar este espacio imagen exhaustivamente. Para imágenes en niveles de gris o en color de tamaños razonables, la misma estimación nos daría cifras insondables. Por lo tanto, parece obvio que el reconocimiento por búsqueda exhaustiva en el espacio imagen es imposible.

Esta representación da lugar al planteamiento de las siguientes preguntas: ¿Cuál es la relación entre los puntos del espacio imagen correspondientes a imágenes de un mismo objeto? ¿Es posible caracterizar eficientemente este subconjunto de puntos de todas las posibles imágenes de un objeto? ¿Puede generarse un modelo de este subconjunto a partir de un conjunto reducido de imágenes (imágenes de entrenamiento)?

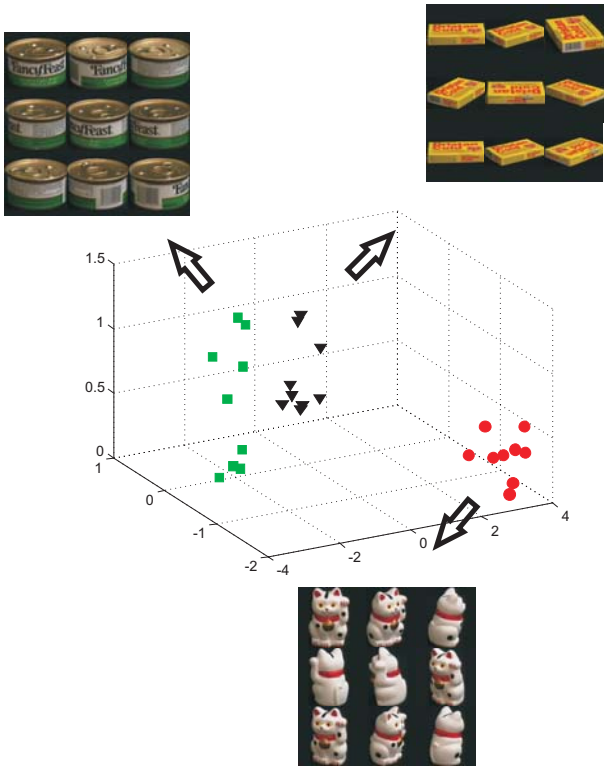


Figura 2.8: Una colección de N imágenes es equivalente a N puntos en el espacio imagen

Consideremos una imagen de un objeto a reconocer, $I(f, c)$, esta imagen es un punto o vector $\mathbf{x}$ en el espacio imagen. Cada uno de los píxeles de la imagen puede ser mapeado en una componente i del punto $(\mathbf{x}_i)$ mediante la relación:

$$i = f * n_c + c \tag{2.2}$$

donde n_c es el número de columnas de la imagen (ancho de la imagen). Una forma directa de reconocer o clasificar una nueva imagen utilizando esta representación consiste en determinar la distancia mínima entre una nueva imagen $\widetilde{\mathbf{x}}$ e imágenes preexistentes en el espacio, $\mathbf{x}$. Dicho de otra forma, dadas K imágenes prototipo de objetos conocidos, cabe encontrar el prototipo que satisfaga:

$$arg\ min_j[d(\widetilde{\mathbf{x}}, \mathbf{x}_k)], \quad k = 1, \ldots, K \tag{2.3}$$

Una distancia métrica a utilizar puede ser la distancia euclídea o norma L2:

$$L2 = d(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\| = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)} = \sqrt{\sum_{i=1}^{fc}(\mathbf{x}_{1i} - \mathbf{x}_{2i})^2} \tag{2.4}$$

La norma L2 es la medida de distancia más utilizada, aunque existen otras medidas tales como la norma L1 u otras versiones de la métrica de Minkowski, que también se utilizan en función de la aplicación [174].

Si todas las imágenes de un mismo objeto se agrupan alrededor de un mismo punto en el espacio imagen, y si ese grupo de puntos están claramente separados del resto de puntos pertenecientes a otros objetos (como en el ejemplo de la figura 2.8), el reconocimiento de objetos se realiza de forma muy simple. En concreto, utilizando el centro de las agrupaciones como prototipos de los objetos a reconocer y una métrica como la norma L2 o la correlación, el problema del reconocimiento de objetos estaría resuelto. Sin embargo, en la práctica no es tan sencillo, especialmente cuando se utilizan imágenes muy grandes, objetos similares o una gran cantidad de objetos. El trabajar con imágenes de considerable tamaño implica que el espacio imagen tendrá una dimensión muy alta, por ejemplo imágenes de resolución 800×600 originarían un espacio imagen de 480000 dimensiones. Trabajar con vectores de tal tamaño es imposible desde el punto de vista computacional para cualquier clasificador, incluso el más sencillo como una medida de distancia. De manera, que resulta necesario utilizar algún método que permita reducir la dimensión del espacio imagen y facilite además la separación entre clases u objetos puesto que esto permitirá una clasificación más sencilla, rápida y robusta.

Los métodos de extracción de características típicos que se utilizan en los sistemas de reconocimiento de patrones, tales como el análisis de componentes principales, el análisis discriminante lineal y el análisis de componentes independientes, entre otros, se pueden aplicar para reducir la dimensión del espacio imagen, o lo que es lo mismo, transformar los datos originales en otros que residen en un espacio de menor dimensión (*subespacio*). Pero el objetivo no es sólo reducir la dimensión del espacio imagen, tambien es necesario separar las clases, es decir, aumentar las diferencias entre los diferentes objetos que se pretende identificar y en este aspecto, el funcionamiento de cada uno de los métodos no es equivalente. Uno de los objetivos de esta tesis es clarificar las ventajas y desventajas que presentan lás técnicas de extracción de características enumeradas en el siguiente capítulo en cuanto a su eficiencia en la tarea de reconocimiento de objetos basada en la apariencia, puesto que la literatura actual no aclara bajo qué condiciones es idóneo cada método.

En el siguiente apartado se presenta una recopilación de los diferentes sistemas de reconocimiento basados en la apariencia que han ido surgiendo a partir de los primeros sistemas propuestos por Turk y Pentland [157], Brunelli y Poggio [28] y Murase y Nayar [118].

2.3. Propuestas previas de sistemas de reconocimiento basados en la apariencia

Los sistemas de reconocimiento basados en la apariencia se han convertido en uno de los métodos más populares para la identificación de objetos y la localización de sus poses en imágenes de niveles de intensidad. El éxito de estos métodos es debido en parte a su capacidad de reconocer sin dificultad objetos de forma libre [3] y manejarse con los problemas de cambios de forma, pose, reflectancia y cambios en la iluminación de los objetos.

La versión más simple de los sistemas de reconocimiento basados en la apariencia consiste en medir distancias directamente sobre el espacio imagen, esta técnica se conoce como *template matching* y sus limitación más importante es la expuesta en la sección anterior: conforme aumenta la resolución de las imágenes, el tamaño del espacio imagen aumenta considerablemente, de forma que el clasificador se ralentiza y aumentan los requerimientos de memoria del sistema. Uno de los primeros trabajos en los que se utiliza la técnica del *template matching* es el de Baron [9] en donde se aplica en la tarea de reconocimiento de caras humanas. Posteriormente, Brunelli y Poggio [28] realizan una comparativa del funcionamiento del *template matching* con los métodos basados en modelo (utilizando vectores de características geométricas de los rostros) también bajo un contexto de reconocimiento de caras humanas. La conclusión de sus resultados es que el utilizar el concepto de espacio imagen mejora las tasas de éxito en el reconocimiento, pero el sistema basado en modelo es más rápido y requiere de menos memoria para almacenar los prototipos de las caras que el sistema de *template matching*.

2.3.1. Sistemas de reconocimiento basados en PCA

La idea básica de utilizar el método de extracción de características clásico, el análisis de componentes principales (PCA), para reducir la dimensión de los vectores en el espacio imagen tiene muchas y diversas variaciones en la literatura que a continuación enumeramos:

- Kirby y Sirovich [91] fueron los pioneros en utilizar PCA como método de compresión de la información en una aplicación de reconocimento visual mediante la apariencia. El concepto básico es utilizar PCA para determinar el mejor sistema de coordenadas (en términos de compresión) para el conjunto de imágenes de entrenamiento. Los vectores base (las componentes principales) del nuevo sistema de coordenadas fueron denominados *eigenpictures*[4].

[3] El concepto de objeto de forma libre se define en el apartado 2.4

[4] En posteriores publicaciones se acuñaron otros términos equivalentes: *eigenfaces*[157] o *eigenimages*[118], según la aplicación.

- Turk y Pentland [157] observaron que al utilizar un sistema de reconocimiento basado en la apariencia, un conjunto reducido de eigenimágenes, esto es, un subespacio de dimensiones reducidas obtenido tras realizar una extracción de características PCA del espacio imagen original, podía ser utilizado para describir y reconstruir caras humanas. Su propuesta fue testeada con una librería de caras humanas compuesta por 16 sujetos cuyas caras habían sido capturadas bajo diferentes iluminaciones, cambios en su pose y escalado (en total se utilizaron 2500 imágenes). Su sistema era capaz de reconocer caras en menos de 1 segundo. Otra aportación importante de su trabajo fue lo que se conoce actualmente como truco de Turk y Pentland, (*Turk and Pentland's trick*), y que consiste en la utilización de un truco algebraico para disminuir el coste computacional del cálculo de los vectores propios de la matriz de covarianzas de los datos. Aunque fue el articulo de Turk y Pentland [157] el que popularizó el empleo de esa técnica para realizar PCA sobre imágenes, el algoritmo fue propuesto originariamente por Murakami y Kumar [117] en 1982. El truco de Turk y Pentland se detalla matemáticamente en el apartado 3.2.1.

- Murase y Nayar [118] desarrollaron el uso del reconocimiento basado en la apariencia para reconocer e identificar la pose de objetos de forma libre. Su sistema de reconocimiento disponía de dos arquitecturas diferentes de subespacios denominadas *universal* y *objeto*, lo que permitía ambas tareas de reconocimiento y localización.

- Nene y Nayar [122] desarrollaron una técnica de búsqueda binaria estructurada para buscar con mayor rapidez dentro de un eigenespacio multidimensional.

- Turk y Pentland [157] discutieron la utilidad de emplear redes neuronales como clasificador en su sistema de reconocimiento de caras basado en PCA. De manera similar, Mukherjee y Nayar [115] experimentaon con redes RBF como clasificador.

- Mundy et al. [116] compararon el sistema basado en la apariencia de Murase y Nayar [118] con dos métodos de reconocimiento basados en modelos geométricos. Su estudio concluye que los sistemas basados en la apariencia proporcionan tasas de éxito en el reconocimiento mayores, pero también tienen un índice elevado de falsos positivos. Esto es debido a la falta de un procedimiento de verificación y a la sensibilidad de los métodos basados en apariencia a oclusiones y errores de segmentación.

- Uno de los principales problemas de los primeros métodos basados en la apariencia era su falta de habilidad para manejar más de un objeto en la escena y/o además en presencia de oclusiones del objeto en la imagen. Huang *et al.* [76] propusieron segmentar las imágenes de entrenamiento de los objetos en partes y utilizar la apariencia de las partes y sus relaciones para identificar los objetos en la escena. Más tarde, Camps *et al.* [30], basándose en la idea previa de [76], ampliaron y mejoraron el sistema basado en partes mediante el empleo de jerarquías entre las relaciones entre las partes de los objetos.

- Otra desventaja de utilizar la representación basada en la apariencia para reconocer los objetos es que éstos han de segmentarse correctamente para que el reconocimiento se realice con éxito. Esto limita el tipo de objetos que pueden ser reconocidos a aquellos que pueden ser segmentados fielmente. Ohba e Ikeuchi [124] y Krumm [94] evitaron este problema mediante la creación de una técnica basada en la apariencia que utilizaba ventanas locales para representar la apariencia del objeto. Ohba e Ikeuchi [124] fueron capaces de trabajar con traslaciones y oclusiones de los objetos utilizando eigenventanas o *eigenwindows*. Las eigenventanas codifican la información de la apariencia de un objeto para una pequeña sección de una de sus vistas. Fueron desarrolladas medidas de detectabilidad, unicidad y fiabilidad para la elección de la eigenventanas y de esta forma, omitir del conjunto de entrenamiento aquellas que fuesen difíciles de detectar o presentasen mucha sensibilidad frente al ruido. Utilizando estas eigenventanas locales fueron capaces de indentificar diferentes objetos en escenarios no controlados. Por otro lado, pero casi simultáneamente, Krumm [94] desarrolló una técnica similar a la de Ohba e Ikeuchi [124]. Las diferencias entre ambas técnicas estriban en la construcción del modelo y el procedimiento de reconocimiento: Krumm utiliza un operador para identificar posibles eigenventanas que no sean redundantes en información.

- Edwards y Murase [50] intentaron resolver el problema de las oclusiones utilizando máscaras para cubrir parte de las eigeimágenes y de la imagen de entrada. Las máscaras se utilizaron dentro de una jerarquía de resoluciones en una búsqueda del objeto de mayor a menor resolución cambiando el tamaño de la máscara.

- Leonardis y Bishop [99] propusieron la selección de puntos (zonas) aleatorios de la escena y sus correspondientes puntos (zonas) en los vectores propios de la transformacion para solucionar el problema de las oclusiones. Su método utilizaba un paradigma de hipótesis y test, donde la hipótesis es un conjunto de localizaciones de los puntos (generadas aleatoriamente) y su eigenespacio prototipo.

- Rao [142] presentó un método en el que utilizaba el aprendizaje adaptativo basado en el principio MDL (*Minimum Description Length*), para calcular los vectores de la transformación PCA. Este enfoque basado en la apariencia de tipo dinámico se utilizó para predecir los cambios temporales y espaciales en la apariencia de una secuencia de imágenes.

- También se han utilizado técnicas basadas en la apariencia para reconocer objetos definidos mediante imágenes de rango o profundidad, como es el caso de Campbell y Flynn [29].

La tabla 2.2 resume las distintas técnicas de reconocimiento enumeradas previamente y que utilizan apariencia y PCA como método de compresión de la información presente en el espacio imagen original.

Método	Oclusiones	Escalado	Tamaño librería	Tasa de Éxito
Kirby y Sirovich [91]	no	sí	-	-
Turk y Pentland [157]	no	no	16	1.0
Murase y Nayar [118] Nayar *et al.* [119]	no	sí	100	1.0
Black and Jepson [24]	no	no	-	-
Camps *et al*. [30]	sí	no	24	1.0
Ohba e Ikeuchi [124]	sí	no	-	-
Krumm [94]	sí	no	2	0.8
Edwards y Murase [50]	sí	sí	6	0.89
Bishop y Leonardis [22]	sí	sí	20	-
Rao [142]	sí	no	-	-
Campbell y Flynn [29]	no	sí	54	0.91

Tabla 2.2: Resumen de los principales trabajos que resuelven el problema del reconocimiento visual de objetos basado en la apariencia utilizando PCA como método de extracción de características.

2.3.2. Sistemas de reconocimiento basados en LDA

Otro método de extracción de características que se ha empleado en los sistemas de reconocimiento de objetos basados en la apariencia es el análisis discriminante lineal (LDA). Esta técnica se ha utilizado con éxito en el reconocimiento de caras [14][52] y en el reconocimiento de objetos de forma libre [103] [155]. Intuitivamente, LDA proporciona un espacio donde los ejemplos visuales de los objetos están más cercanos entre sí y a la vez más separados del resto de objetos del sistema. Como se mostrará en el capítulo siguiente, la técnica de LDA es superior al método PCA si el número de las imágenes de entrenamiento son suficientemente representantivas de todo el rango de variación de las vistas del objeto [111]; en otro caso, la diferencia de los resultados entre ambos métodos es insignificante.

2.3.3. Sistemas basados en ICA

En los últimos años un número considerable de estudios experimentales están imponiendo el uso del análisis de componentes independientes (ICA) como método de extracción de características en los sistemas de reconocimiento de objetos basados en la apariencia. En la mayoría de estas propuestas se realizan comparativas entre el funcionamiento de PCA e ICA bajo el mismo contexto de objetos a reconocer, y aunque en la mayor parte de los casos las tasas de éxito en el reconocimiento obtenidas presentan al método ICA como el vencedor, puesto que son ligeramente superiores a las de PCA, (como es el caso de Bartlett *et al*., [12][10][11], Donato *et al*., [44], Fortuna *et al.* [61], Liu y Wechsler [102][104][101], Yuen y Lai [178], Draper *et al*. [47], Sahambi y Khorasani [144], Eke-

Autores	Mejores resultados	Algoritmo ICA utilizado
Bartlett and Sejnowski, 1997 [10]	ICA	Infomax
Bartlett *et al.*, 1998 [12]	ICA	Infomax
Donato *et al.*, 1999 [44]	ICA	Infomax
Liu and Wechsler, 1999 [102]	ICA	Comon's
Yuen and Lai, 2000 [178]	ICA	Infomax
Baek *et al.*, 2002 [7]	PCA	Infomax
Bartlett *et al.*, 2002 [11]	ICA	Infomax
Fortuna *et al.*, 2002 [61]	ICA	FastICA
Moghaddam, 2002 [114]	Ambos	JADE y FastICA
Socolinsky and Selinger, 2002 [149]	Ambos	FastICA
Draper *et al.*, 2003 [47]	ICA	Infomax
Liu and Wechsler, 2003 [104]	ICA	Comon's
Sahambi and Khorasani, 2003 [144]	ICA	Infomax
Liu, 2004 [101]	ICA	Comon's
Ekenel and Sankur, 2005 [51]	ICA	FastICA
Martiriggiano, 2005 [110]	ICA	FastICA

Tabla 2.3: Principales trabajos que realizan comparativas del funcionamiento de PCA e ICA como método de extracción de características en tareas de reconocimiento visual de objetos basado en la apariencia.

nel y Sankur [51] y Martiriggiano *et al*. [110]); otros autores proporcionan resultados contradictorios: Baek *et al.* [7] afirma la superioridad de PCA y Moghaddam [113] concluye que no existen diferencias experimentales estadísticas entre los dos métodos, por lo que su funcionamiento en tareas de reconocimiento debe ser equivalente. En un estudio con imágenes en luz visible e infrarojos, Socolinsky y Salinger [149] concluyen que ICA supera a PCA en imágenes obtenidas con luz visible, pero que PCA funciona mejor que ICA en imágenes LWIR[5].

La tabla 2.3 presenta una recopilación de los principales estudios que abordan este problema de comparar el funcionamiento de ambos métodos bajo el marco del reconocimiento de objetos. Todos los trabajos que ahí se enumeran presentan resultados experimentales obtenidos con clasificadores tipo vecino más cercano y utilizando distancias métricas estandar como la norma L2, la correlación o el coseno. Distintos autores utilizan algoritmos de ICA diferentes, lo que no ayuda a aclarar la confusión entre los resultados obtenidos. En ningún caso se estudia el problema teóricamente comparando ambos procesos.

Uno de los objetivos de esta tesis es el averiguar el porqué de estos resultados experimentales contradictorios. Pero para ello se abordará el problema bajo el punto de vista

[5]Long-Wave Infrared

analítico, mediante la comparación de los procesos que implican cada uno de los principales métodos.

2.3.4. Sistemas basados en otras transformaciones

Otro método de extracción de características de reciente creación es la factorización de matrices no negativas (NMF) [129] [96], que intenta obtener una representación no negativa de los datos del espacio imagen original. Se trata de un algoritmo de inspiración neurobiológica basado en el hecho que las respuestas neuronales no pueden ser negativas: sólo positivas o nulas. El éxito de NMF radica en su capacidad de obtención de características locales a partir de los datos originales, esto es, los vectores del subespacio NMF resaltan rasgos locales de los objetos de las imágenes originales. En concreto, Lee y Seung [96] observaron esta representación local al utilizar su algoritmo con las imágenes de caras humanas de la librería CBCL [2]. La figura 2.9(a) muestra el conjunto de imágenes originales y la figura 2.9(b) la representación local obtenida con el algoritmo de NMF de Lee y Seung.

A partir de los resultados presentados por [96], han surgido numerosos trabajos que proponen la utilización del algoritmo NMF original o alguna modificación de éste [97], para el cómputo de los vectores del subespacio que consigan la reducción de la dimensión de las imágenes del espacio imagen en los sistemas de reconocimiento basados en la apariencia.

Guillamet *et al.* en [68] [69] proponen una variante del algoritmo, WNMF (Weighted Non-negative Matrix Factorization), que pretende mejorar las capacidades de representación local y lo emplean exitosamente en la tarea de identificación de objetos utilizando un clasificador que combina los resultados de PCA y WNMF obtenidos a partir de los histogramas de color de un conjunto de objetos formado por las portadas a color de revistas y periódicos.

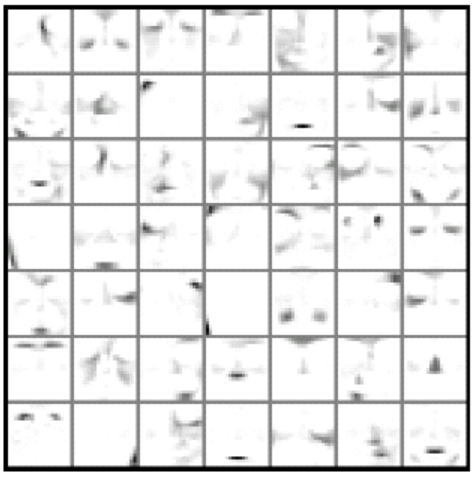
Otros trabajos como los presentados por Feng *et al.*[54] y Rajapakse yWyse, [140][141], presentan comparativas entre el funcionamiento de PCA, ICA y NMF, obteniendo resultados confusos, pues las tasas de reconocimiento más altas son proporcionadas por uno u otro método según el conjunto de datos utilizado o las medidas de distancia empleadas.

Otro de los objetivos de esta tesis es el estudio experimental del funcionamiento de un nuevo algoritmo de NMF que permite obtener siempre representaciones locales de los objetos, independientemente de su naturaleza, forzando el nivel de *sparseness*[6] de los vectores del subespacio. Se trata del algoritmo NMFSC (Non-negative Matrix Factorization with Sparseness Constraints)[75]. Con este nuevo algoritmo se consigue obtener una

[6]El concepto de *sparseness* se refiere a un tipo específico de distribución estadística que cumple la representación de datos utilizada, y se describe con más detalle en el apartado 3.4.3.

representación que puede ser beneficiosa para el reconocimiento de los objetos si existen oclusiones parciales de los mismos.

(a)



(b)

Figura 2.9: La figura 2.9(a) muestra el conjunto de imágenes originales de caras humanas de la librería CBCL [2] y la figura 2.9(b) los vectores NMF obtenidos con el algoritmo de Lee y Seung[96].

2.4. Objetos de forma libre y sus librerías

El principal éxito de los sistemas de reconocimiento basados en la apariencia es su capacidad de reconocer *objetos de forma libre*. No existe una definición formal o exacta del término *objeto de forma libre* o *free-form object*[7]. En general, con el término *libre* se intenta caracterizar al conjunto de los objetos cuyas superficies no pueden clasificarse fácilmente como planas o cuadráticas. Por lo tanto, un objeto de forma libre se asume que está formado por superfícies no-planas y no-cuadráticas (*free-form surface*[8]). Dorai y Jain [46], Besl [18] y Stein y Medioni [152] citan como ejemplos típicos de objetos libres de forma a esculturas, coches, cascos de barcos, aviones, caras humanas, órganos y mapas terrestres, entre otros muchos. Los objectos estadísticos que definen tipos de textura, o aquellos que poseen características de auto-similitud y que se describen utilizando modelos de fractales, o superficies no-orientables como por ejemplo la cinta de Moebius o la botella de Kein son excluidos de la categoría de *objeto de forma libre*.

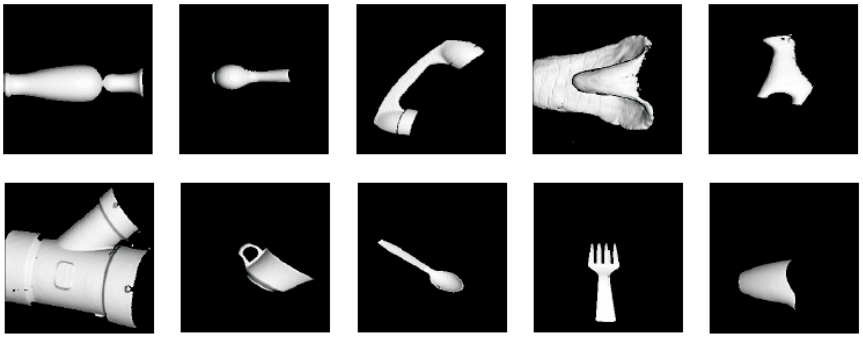


Figura 2.10: Varias vistas de imágenes de rango de *objetos de forma libre* artificiales presentados por Dorai y Jain[46].

De manera intuitiva, es posible definir un objeto de forma libre como aquél que no se aproxima a un cuerpo geométrico simple (no es un geón básico[19]) y que puede existir en la naturaleza. Con esta sencilla definición, queda claro que la mayoría de los objetos reales pertenecen a la categoría de objetos de forma libre.

Existen librerías de imágenes de objetos de forma libre que son utilizadas ampliamente para testear el funcionamiento de los nuevos métodos de reconocimiento de objetos que se plantean. Entre estas librerías destaca, por ser la más utilizada y por su considerable tamaño, la librería COIL-100 (*Columbia Object Image Library*) [120], compuesta por

[7] Otros sinónimos utilizados en la literatura son *sculpted*, *free-flowing* o *piecewise-smooth object*

[8] Una definición fue propuesta por Besl[18]: *A free-form surface has a well defined surface normal that is continuos almost everywhere except at vertices, edges and cups*.

imágenes de color de 100 objetos de forma libre. La figura 2.12 muestra los 100 objetos que la componen. Están disponibles 72 vistas diferentes de cada uno de los objetos, obtenidas mediante un dispositivo similar al que se muestra en la figura 2.11, compuesto por una cámara y una mesa rotatoria (360°), que permite la adquisición de diferentes vistas del objeto cada 5°.

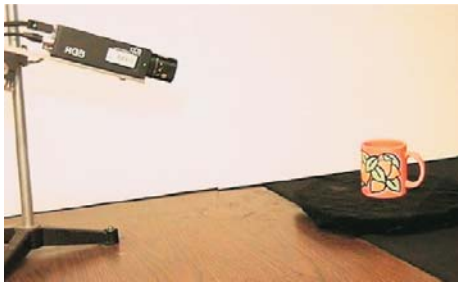


Figura 2.11: Dispositivo compuesto por una mesa giratoria y una cámara para la adquisición de las imágenes de los objetos de la librería COIL-100.

Figura 2.12: Conjunto de los objetos que componen la librería de imágenes de objetos de forma libre COIL-100

Recientemente ha sido creada una librería de objetos inspirada en la COIL-100 pero de considerable mayor tamaño, se trata de la colección ALOI (*Amsterdam Library of Object Images*)[65]. ALOI está compuesta por las imágenes de 1000 objetos tomadas

en muy diversas condiciones: 72 vistas tomadas con una separación de 5° (de manera similar a la COIL-100), 24 imágenes con diferentes ángulos de iluminación y 12 imágenes bajo colores de iluminación diferentes. Adicionalmente dispone de imágenes adquiridas por un par estéreo de unos 750 objetos. La colección ALOI contiene un total de 110250 imágenes. Las figuras 2.13 a 2.15 ilustran el tipo de imágenes que dispone la librería ALOI para cada uno de sus 1000 objetos.

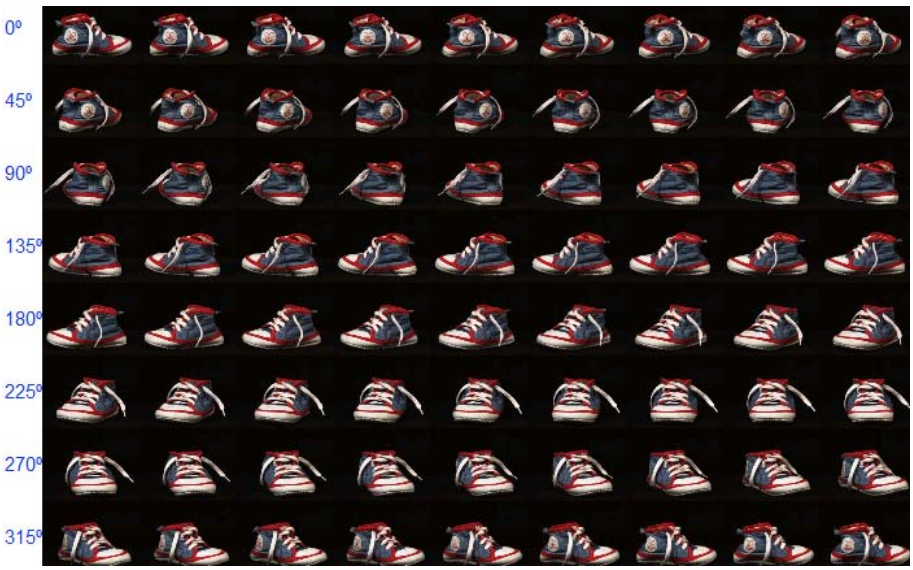
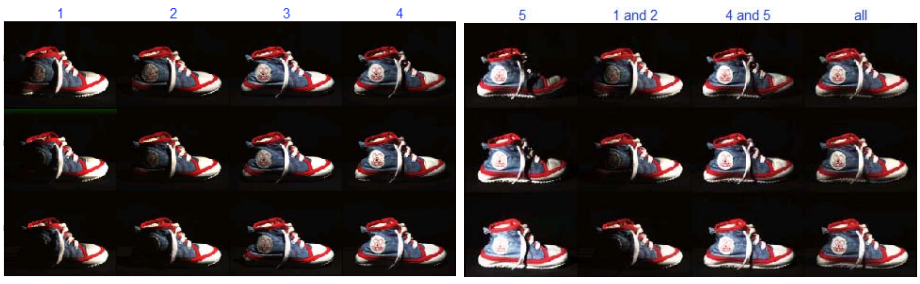


Figura 2.13: Ejemplo de las imágenes de un objeto de la librería ALOI visto bajo 72 ángulos distintos (paso de 5°).



(a) (b)

Figura 2.14: Ejemplo de las imágenes de un objeto de la librería ALOI visto bajo 24 direcciones de iluminación diferentes.

Otra librería interesante es la SOIL-47 (*Surrey Object Image Library*)[93], compuesta por 47 objetos en diferentes vistas y grados de iluminación. Además dispone de una co-

Figura 2.15: Ejemplo de las imágenes de un objeto de la librería ALOI visto bajo 12 colores de iluminación (temperaturas) diferentes.

lección de escenas complejas donde existe más de un objeto, siendo útiles para el test de algoritmos de detección de objetos.

Figura 2.16: Algunos ejemplos de escenas complejas con objetos de la librería SOIL-47.

La *View Sphere Database* [70] contiene una colección extensa de vistas de 8 objetos diferentes. Las vistas han sido adquiridas siguiendo la superfície de una esfera geodésica, por lo tanto, todas las imágenes están a la misma distancia del objeto. Dispone de dos conjuntos de vistas para cada objeto: uno que agrupa 91 vistas de la semiesfera superior y 172 de la esfera completa, y otro conjunto (de test) de 97 vistas adquiridas desde la misma esfera pero en diferentes condiciones.

Y por ultimo, cabe destacar la librería ETH-80, diseñada para el test de algoritmos de catalogación, puesto que está formada por imágenes de 80 objetos pertenecientes a 8 categorías: manzanas, peras, tomates, vacas, perros, caballos, tazas y coches. Además proporciona la máscara de segmentación del objeto de cada una de las imágenes. La figura 2.18 muestra una imagen de cada uno de estos 80 objetos.

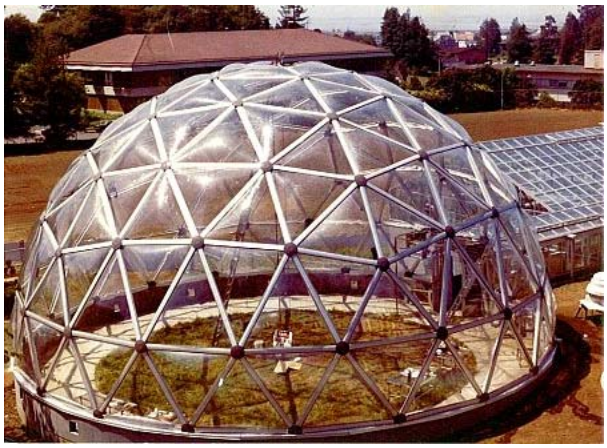


Figura 2.17: Esfera geodésica.

Figura 2.18: Conjunto de los objetos que componen la librería de imágenes de objetos de forma libre ETH-80

2.5. Conclusiones

En este capítulo se han descrito los principales métodos de reconocimiento de objetos que existen en la actualidad, haciendo especial hincapié en la distinción entre sistemas de reconocimiento basados en modelo y basados en apariencia. La presente tesis se centra en el reconocimiento de objetos basado en la apariencia.

A lo largo de este capítulo se han enumerado, según el método de extracción de características empleado, los principales estudios previos que utilizan este tipo de reconocimiento. En la actualidad, no existe una postura común acerca de cúal es el método de extracción de características idóneo a utilizar en los sistemas basados en apariencia. Uno de los objetivos de la presente tesis es clarificar las contradicciones existentes en cuanto a los resultados de reconocimiento de los métodos PCA/ICA, así como justificar el contexto bajo el cuál otros métodos nuevos, como RP[9] y NMF pueden aportar ventajas en el reconocimiento de los objetos. En el siguiente capítulo se describen los fundamentos teóricos de los diferentes métodos de extracción de características que se aquí se han mencionado.

[9]La técnica de *Random Projection* (RP) se describe con detalle en el siguiente capítulo.

Capítulo 3

Técnicas de extracción de características para los métodos de reconocimiento basados en la apariencia

Uno de los problemas centrales en el campo del reconocimiento de objetos basado en la apariencia es la búsqueda de una representación adecuada de los datos, a través de una transformación de los mismos. Esto es importante para el posterior analisis de los datos y el buen funcionamiento del sistema de reconocimiento. Y además, es uno de los objetivos de la presente tesis: la búsqueda de la representación óptima de un conjunto de variables multidimensionales discretas obtenidas a partir de las imágenes de los objetos a reconocer.

Se han desarrollado diversos principios y métodos para encontrar transformaciones adecuadas. Entre estos, destacan el análisis de componentes principales, el análisis de Fisher, el análisis de componentes independientes, la factorización de matrices no negativas y el método de las proyecciones aleatorias. En general, estos métodos definen un principio que justifica cúal es la transformación óptima. Esta transformación puede definirse en el sentido de una reducción de la dimensionalidad de los datos óptima, un cierto interés estadístico de las componentes resultantes, simplicidad en la transformación u otros criterios, incluyendo aquellos orientados a aplicaciones concretas.

No existe acuerdo entre las investigaciones previas acerca de cuáles son las mejores transformaciones para una aplicación de reconocimiento de objetos; ni en cuanto a los resultados experimentales ni en cuanto a los fundamentos teóricos capaces de explicar tales resultados. Uno de los objetivos de la presente tesis consiste en aportar algo de luz ante las aparentes contradicciones de trabajos previos e intenta justificar los resultados obtenidos en tales trabajos a partir de un estudio teórico riguroso del comportamiento de las diferentes transformaciones.

En el presente capítulo se repasan los principios fundamentales de algunas de estas transformaciones, pero antes se presenta el esquema básico de un sistema de reconocimiento de patrones para luego seguir con la descripción detallada de algunas de las tecnicas de extraccion de caracteristicas más utilizadas y cuyo funcionamiento se pretende analizar en el marco de los sistemas de reconocimiento de objetos basados en la apariencia.

3.1. Descripción de los procesos básicos que constituyen los sistemas de reconocimiento de patrones

El esquema general de un sistema de reconocimiento de patrones se muestra en la figura 3.1. A partir de un patrón natural (en nuestro caso, un objeto tridimensional), un bloque receptor adquiere información del patrón (*la cámara capta la imagen del objeto*). El bloque de extracción de características debe extraer la información relevante de la totalidad de los datos captados por el sensor en la fase anterior (*color del objeto en la imagen, silueta, textura,..*). Y por último el módulo clasificador ha de etiquetar de forma automática el patrón (*identificar el objeto captado con la cámara: es una cara, un objeto, un paisaje,* ...).

Este esquema no debe verse como los pasos a seguir en la construcción de un sistema de reconocimiento de patrones, sino más bien desde un punto de vista funcional: la entrada es un patrón natural y el resultado es una etiqueta.

El esquema de la figura 3.1 corresponde a una configuración estadística de reconocimiento de patrones. La configuración estadística es la más simple y consiste en representar cada patrón mediante un vector de números, resultantes del muestreo y cuantificación de las señales externas y cada clase por uno o varios patrones prototipo. El reconocimiento de patrones estadístico no considera el contexto del patrón a etiquetar, es decir, la relación entre diferentes patrones. También podría contemplarse que los patrones complejos pudieran descomponerse recursivamente en patrones más simples hasta llegar a componentes básicos. El sistema de reconocimiento que hace uso de esta información de contexto se suele denominar sintáctico. En el reconocimiento sintáctico, cada patrón se describe en términos de sus elementos básicos (elementos terminales) y unas reglas sintácticas (gramática). El reconocimiento de voz es un ejemplo de configuración de reconocimiento sintáctica. El reconocimiento de objetos mediante visión por computador se enfoca normalmente desde el punto de vista estadístico, aunque existen excepciones a este enfoque y se han desarrollado algunas teorías que pretenden resolver el problema de la representación y reconocimiento de objetos desde el punto de vista sintáctico; dentro de estas teorías cabe destacar la teoría de los geones desarrollada por [19] y comentada en el capítulo anterior.

A continuación se definirá de manera más formal cada uno de los elementos que forman parte de un sistema de reconocimiento de patrones estadístico:

3.1.1. Módulo de adquisición de datos

La entrada a un sistema de reconocimiento de patrones estadístico es un vector numérico que contiene los valores muestreados y cuantificados de una serie de señales naturales. De una manera más formal, suponiendo patrones M-dimensionales, un patrón $\mathbf{x}$ es una

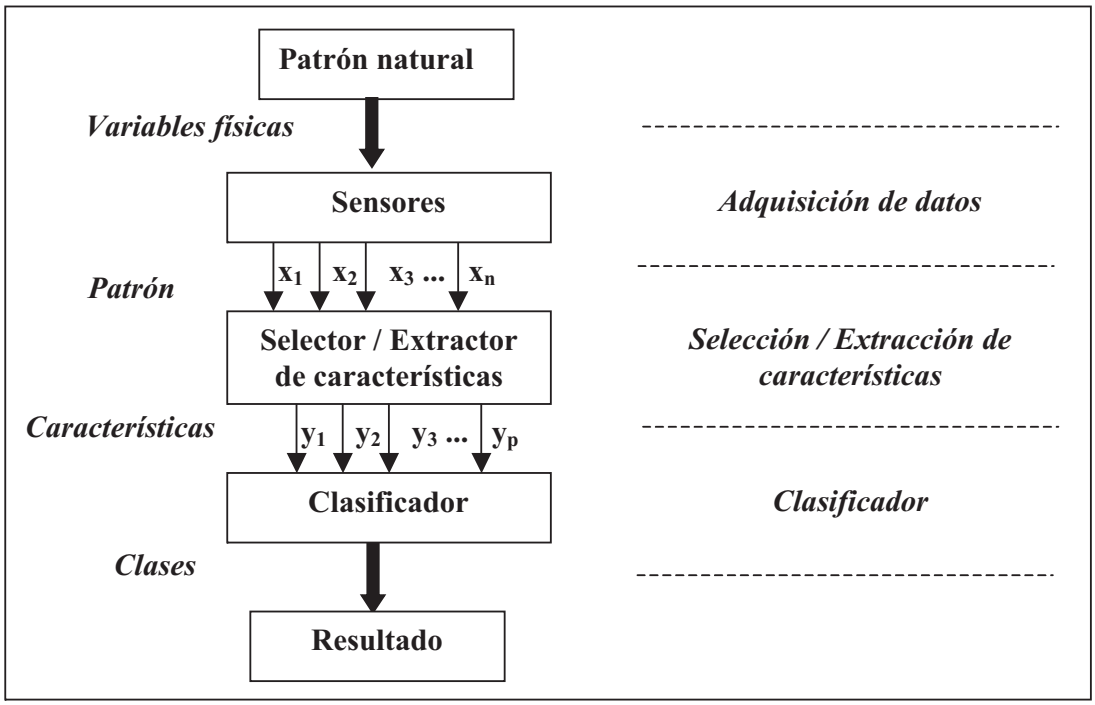


Figura 3.1: Esquema modular de un sistema de reconocimiento de patrones.

variable aleatoria M-dimensional compuesta por M componentes, $x_1, x_2, ..., x_M$:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \tag{3.1}$$

Con esta aproximación un patrón no es más que un punto en el espacio de representación de los patrones, que es un espacio de dimensión determinada por el número de variables consideradas. El espacio de patrones $\mathbf{X}$ se define como el conjunto de todos los valores posibles (N) que puede tomar cada patrón $\mathbf{x}$.

3.1.2. Módulo de selección y extracción de características

En un problema de clasificación, a la hora de tomar decisiones, los humanos consideramos tan sólo unos pocos atributos o variables de entre todos los posibles. El mecanismo exacto que utiliza el cerebro para seleccionar la información relevante es desconocido; pero, es evidente que un buen conjunto de atributos debería potenciar la similaridad entre patrones de la misma clase y la separación entre patrones de clases diferentes. Cuanto más complejos son los patrones que se manejan, más difícil resulta dar una explicación

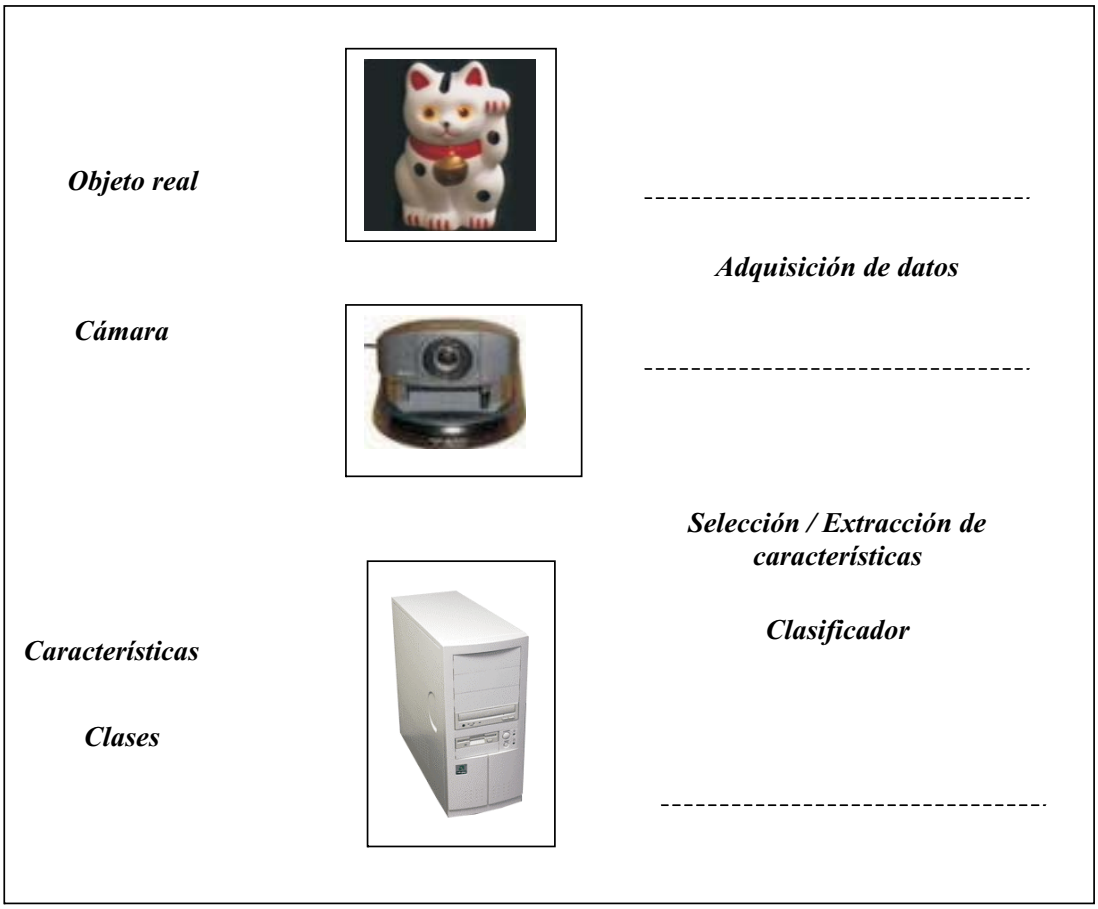


Figura 3.2: Esquema modular de un sistema de reconocimiento de objetos mediante imágenes bidimensionales.

formal y calculable al problema de explicitar la bondad de estos atributos. Una solución indirecta a esta cuestión consiste en utilizar todos los atributos que pudieran aportar información relevante al problema que se trata de resolver, pensando que a falta de conocer la calidad, la cantidad puede ayudar. No obstante, una cantidad importante de características aumenta la complejidad del clasificador, a este hecho se le conoce en la literatura como la maldición de la dimensión [48] (*curse of dimensionality*). Este término es debido a Bellman [17] y, en estadística, se refiere al hecho de que la convergencia de cualquier estimador al valor real de la función de probabilidad definida en un espacio de muy alta dimensionalidad es muy lenta. En términos de reconocimiento y detección de objetos significa que, a priori, se necesita una cantidad enorme de observaciones de un objeto para obtener un buen estimador de la función que identifique correctamente al objeto. Para resolver el problema de la maldición de la dimensión, la estrategia a seguir es reducir el tamaño del espacio de características.

De todo esto, puede concluirse que antes de proceder a la construcción del clasificador se requiere un estudio detallado del conjunto de atributos disponibles para proceder a la selección de los más relevantes o para derivar nuevos atributos con un significado más claro. Este paso previo a la construcción del clasificador se conoce con el nombre genérico de selección y extracción de características, y puede definirse como el proceso que selecciona un conjunto reducido y significativo de características (variables o atributos).

La selección y extracción de características trata de resolver el problema de extraer la información más relevante para la clasificación entre la suministrada por los sensores (datos en bruto). Formalmente esto se puede plantear del siguiente modo, dado un conjunto de patrones M-dimensionales:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \tag{3.2}$$

se trata de obtener un nuevo conjunto (características) P-dimensionales :

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{pmatrix} \tag{3.3}$$

donde $P \leq M$.

Matricialmente, dado un conjunto $\mathbf{X}$ de N patrones M-dimensionales , $\mathbf{X} \in \Re^{M \times N}$, se trata de obtener un nuevo conjunto $\mathbf{Y}$, $\mathbf{Y} \in \Re^{P \times N}$, donde $P \leq M$.

La tarea que realizará el módulo de selección y extracción de características será la de extraer la información más relevante a partir de toda la información original y representarla en un espacio de menor dimensión cuyas componentes contengan la mayor cantidad de información. Es posible distinguir dos tareas diferenciadas, aunque en ocasiones muy relacionadas: *selección* y *extracción* de características.

3.1.2.1. Selección de características

Estas técnicas resultan apropiadas cuando el coste de la adquisición y manipulación de todas las medidas disponibles es muy elevado o cuando no resultan aplicables por razones teóricas.

Así, si $\mathbf{X}$ es el conjunto de los N vectores originales M-dimensionales, sea $\mathbf{Y}$ un subconjunto escogido de dimensión P tal que ($P < M$). De forma que para cada patrón $\mathbf{x}$:

$$\begin{aligned} \mathbf{x} &= \{x^i | i = 1, 2, ..., M\} \\ \mathbf{y} &= \{y^j | j = 1, 2, ..., P, \;\; y^j \in \mathbf{x}\} \end{aligned} \tag{3.4}$$

El conjunto $\mathbf{y}$ es un conjunto reducido de variables seleccionadas de $\mathbf{x}$. Matemáticamente, la selección de características obtiene un conjunto de P variables de un conjunto inicial de M variables optimizando una función criterio J. Esta función criterio será una medida de separabilidad, funciones criterio que se utilizan con frecuencia son la distancia de Jeffries-Matusita media y la divergencia transformada media [48] .

Una vez fijada la función criterio, sólo falta especificar el método de optimización que se utilizará. De acuerdo a la optimalidad de la solución proporcionada estos métodos se clasifican en:

- Algoritmos de búsqueda óptima: proporcionan una solución óptima, pero a costa de un gran esfuerzo computacional.
- Algoritmos de búsqueda subóptima: proporcionan una buena solución, no necesariamente la mejor, pero con un menor coste computacional.

Otra clasificación de los algoritmos de selección de características, de acuerdo a la filosofía subyacente en su contrucción es la siguiente:

- Técnicas ”bottom up”: parten de un conjunto vacío de variables al que se van añadiendo nuevas variables gradualmente.
- Técnicas ”top-down”: parten del conjunto original (completo) de variables del que se van eliminando variables gradualmente.

La tabla 3.1 muestra un listado de los principales métodos de selección de características que se han propuesto en la literatura [85].

3.1.2.2. Extracción de características

Se habla de técnicas de extracción de características cuando la finalidad es la de transformar el espacio de patrones original, $\mathbf{X}$, en un nuevo espacio $\mathbf{Y}$ que no es necesariamente un subespacio de $\mathbf{X}$, a diferencia de las técnicas de selección de características. Para esta transformación se usa toda la información del espacio original, $\mathbf{X}$, para obtener nuevos vectores derivados de características, $\mathbf{Y}$. Algunas de las nuevas variables pueden incluso eliminarse de forma directa, proporcionando, adicionalmente, un mecanismo de selección de características. Por ejemplo, podrían utilizarse estas técnicas para comprimir la información en un espacio de menor dimensionalidad para facilitar su transmisión o almacenamiento.

Métodos óptimos	**Fundamento**
Búsqueda exhaustiva o por fuerza bruta	Evaluar todos los $\binom{M}{P}$ posibles subconjuntos
Ramificación y poda (*Branch and bound*)	Algoritmo top-down con la posibilidad de vuelta atrás, lo que permite examinar implícitamente todas las combinaciones
Métodos subóptimos	**Fundamento**
Selección incremental secuencial *Secuential Forward Selection (SFS)*	Algoritmo bottom-up que selecciona una variable para su adición en cada paso
Selección decremental secuencial *Secuential Backward Selection (SBS)*	Algoritmo top-down que selecciona una variable para su eliminación en cada paso
Algoritmo (l, r)	Primero aumenta el conjunto con l características utilizando SFS y luego elimina r características con SBS
Algoritmo de selección paso a paso (*stepwise Jeffries-Matusita* *feature selection algorithm*)	Procesamiento iterativo con vuelta atrás basado en 2 pasos: inclusión y exclusión

Tabla 3.1: Principales métodos de selección de características

Podemos expresar la transformación mediante una función W tal que:

$$\mathbf{Y} = W(\mathbf{X}) \tag{3.5}$$

y la forma concreta de la transformación W se obtiene optimizando alguna función criterio $J(\Delta)$. Concretamente, se tratará de escoger entre todas las transformaciones $\widetilde{W}(\mathbf{X})$ la transformación $W(\mathbf{X})$ como la que satisface

$$J(W(\mathbf{X})) = max \;\; J(\widetilde{W}(\mathbf{X})) \tag{3.6}$$

En general, la transformación $W(\mathbf{X})$ puede ser de cualquier tipo, aunque en la práctica, la mayoría de las técnicas de extracción de características se restringen a funciones lineales, esto es,

$$\mathbf{Y} = \mathbf{W}^T\mathbf{X} \tag{3.7}$$

donde $\mathbf{W}$ es una matriz de coeficientes de transformación de orden $M \times P$.

Algunas de las técnicas de extracción de características más utilizadas se enumeran en la tabla 3.2. En apartados posteriores se expondrán con más detalle algunos de estos métodos de extracción de características que son parte fundamental en el proceso de reconocimiento de objetos basado en la apariencia. En concreto, el Análisis de Componentes Principales (PCA), el Análisis Discriminante Lineal (LDA), la Factorización de Matrices

Métodos	Características
Análisis de componentes principales *Principal Component Analysis* (PCA)	Transformación lineal, no supervisada, basada en la descomposición en vectores propios
Análisis Discriminante Lineal *Linear Discriminant Analysis* (LDA)	Transformación lineal, supervisada, basada en la descomposición en vectores propios
Búsqueda de proyecciones *Projection Pursuit*	Transformación lineal, iterativa, no gaussiana
Kernel PCA	Transformación no lineal, no supervisada basada en la descomposición en vectores propios
PCA basado en redes neuronales	Transformación lineal, iterativa
Redes Autoasociativas No Lineales	Transformación no lineal, iterativa, no gaussiana
Análisis de Componentes Independientes *Independent Component Analysis* (ICA)	Transformación lineal, iterativa, no gaussiana
Factorización de Matrices No Negativas *Non Negative Matrix Factorization* (NMF)	Transformación lineal, iterativa, no gaussiana

Tabla 3.2: Principales métodos de extracción de características

No-negativas (NMF), el Análisis de Componentes Independientes (ICA) y el Método de las Proyecciones Aleatorias (RP).

3.1.3. Módulo de clasificación: aprendizaje y entrenamiento

3.1.3.1. Conceptos generales

El objetivo final de un sistema de reconocimiento de patrones es el etiquetar de forma automática patrones de los cuales desconocemos su clase. Suponemos que el sistema dispone de un módulo de adquisición de datos y que se han seleccionado previamente las variables más significativas, cabe ahora definir cuál es el conjunto de etiquetas o dicho de otra forma qué salidas debe proporcionar el sistema. El conjunto de todas las posibles etiquetas se denomina como el conjunto de las clases.

Si suponemos que todos los patrones a reconocer son elementos potenciales de J clases distintas denotadas ω_j, $j = 1, 2, ..., J$, llamaremos

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_J\} \tag{3.8}$$

al conjunto de las clases.

Resulta conveniente ampliar el conjunto incorporando una nueva clase, llamada la clase de rechazo. Así, se define la clase de rechazo (ω_0) como una clase que se asigna

a todos los patrones para los que no se tiene una certeza aceptable de ser clasificados correctamente en alguna de las clases de Ω.

$$\Omega^* = \{\omega_1, \omega_2, \ldots, \omega_J, \omega_0\} \tag{3.9}$$

Una vez establecido el conjunto de clases se procede al diseño del clasificador. Como puede intuirse, el diseño y construcción del clasificador no es una tarea trivial ni directa, y de su buen funcionamiento depende todo el sistema de reconocimiento. El clasificador es por tanto la etapa más importante dentro del sistema de reconocimiento.

Un clasificador o regla de clasificación es una función $f : \mathbf{X} \rightarrow \Omega^*$ definida sobre los patrones $\mathbf{x}$ tal que para todo patrón $\mathbf{x}$, $f(\mathbf{x}) \in \Omega^*$. Los términos aprendizaje y entrenamiento se utilizan para referirse al proceso de construcción del clasificador. El aprendizaje puede realizarse de dos maneras muy diferentes: aprendizaje supervisado y no supervisado.

Un aprendizaje supervisado requiere disponer de un conjunto de patrones de los cuales se conoce su clase cierta. A este conjunto se le denomina conjunto de entrenamiento. Este tipo de entrenamiento se denomina entrenamiento supervisado y los clasificadores así obtenidos clasificadores supervisados. Disponer de un conjunto de entrenamiento supone que alguien se ha preocupado de etiquetar los patrones de ese conjunto.

El aprendizaje no supervisado se realiza a partir de un conjunto de patrones del que no se conoce su clase cierta. En ocasiones, ni siquiera se conoce el número de clases. Básicamente, se traduce en encontrar agrupamientos. El objetivo suele ser el de verificar la validez del conjunto de clases para una clasificación supervisada. Las técnicas utilizadas suelen denominarse métodos de agrupamiento o clustering.

Parece evidente que el reconocimiento de objetos reales tridimensionales a partir de imágenes bidimensionales cabe enfocarlo dentro de un marco de aprendizaje supervisado, puesto que sí se dispone de un conjunto de patrones de los que se conoce su clase cierta. Las técnicas no supervisadas están destinadas a aplicaciones de reconocimiento más complejas donde se desconocen las clases de los patrones e incluso a veces no se sabe a priori cuántas clases pueden existir.

En el aprendizaje supervisado, si consideramos que en un caso ideal cada agrupamiento representa a una clase y cada clase tiene asociado un agrupamiento bien diferenciado de los demás, un problema de clasificación supervisada puede plantearse como la búsqueda de las superficies que separan los diferentes agrupamientos. Estas superficies se denominan superficies de decisión. Las superficies de decisión determinan regiones de decisión de forma que cada clase tiene asociada una región en el espacio de características y la decisión sobre la clase a asignar a un nuevo patrón se hará en base a la región en la que

éste se encuentra en el espacio.

La búsqueda de estas superficies (análogamente, regiones) de decisión se puede abordar de dos maneras, dependiendo de si se conoce o supone un determinado modelo estadístico para las clases. Si se supone un completo conocimiento a priori de la estructura estadística de las clases, el aprendizaje se reduce a la estimación de los parámetros que determinan las funciones de densidad de probabilidad de las clases. Las fronteras de decisión están definidas por las distribuciones de probabilidad de las clases. Los clasificadores construidos bajo esta suposición se conocen como clasificadores paramétricos.

Si no se supone un determinado modelo estadístico, bien por desconocimiento o por la imposibilidad de asumir un modelo paramétrico adecuado, el problema resulta más complejo y se puede abordar desde diferentes perspectivas. Las fronteras de decisión están definidas por los prototipos. Los clasificadores construidos sin esta suposición se conocen como clasificadores no paramétricos.

Bajo este punto de vista, para construir el clasificador, en primer lugar debe estudiarse cómo se distribuyen los patrones en cada clase (la estructura estadística de las clases) y establecer una función discriminante adecuada utilizando los prototipos disponibles. En esto consiste básicamente el entrenamiento supervisado paramétrico. En el caso paramétrico, el reconocimiento de patrones puede verse, desde un punto de vista muy general, como un problema de estimación de funciones de densidad de probabilidad en un espacio multidimensional para posteriormente dividir el espacio en regiones de decisión asociadas a clases. La herramienta fundamental para esta tarea es la Estadística y particularmente la Teoría de la decisión de Bayes.

En la teoría bayesiana, un patrón $\mathbf{x}$ perteneciente a una clase determinada ω_j es considerado una observación aleatoria de la función de densidad de probabilidad $p(\mathbf{x}|\omega_j)$. Existen diferentes reglas de decisión para definir las fronteras o regiones de decisión entre clases. Entre todas ellas, las más extendidas son la regla de decisión de Bayes y la regla de máxima verosimilitud, que es un caso particular de la primera. La regla de decisión bayesiana óptima para minimizar el riesgo condicionado (la pérdida esperada asociada a la decisión) puede establecerse de la siguiente manera: asignar un patrón de entrada $\mathbf{x}$ a la clase ω_j para la cual el riesgo condicionado

$$R(\omega_j|\mathbf{x}) = \sum_{i=1}^{c} L(\omega_i|\omega_j) \cdot P(\omega_j|\mathbf{x}) \tag{3.10}$$

es mínimo. En la ecuación (3.10), $L(\omega_i|\omega_j)$ es la pérdida que se comete al decidir la clase ω_i cuando la clase cierta es ω_j, y $P(\omega_j|\mathbf{x})$ es la probabilidad a posteriori [48]. En el caso de la función de pérdida $0/1$, definida en (3.11), el riesgo condicionado se convierte en la probabilidad condicionada de error en la clasificación, tal como se muestra en la ecuación (3.12).

$$L(\omega_i|\omega_j) = \begin{cases} 0, i = j \\ 1, i \neq j \end{cases} \tag{3.11}$$

$$R(\omega_j|\mathbf{x}) = 1 - P(\omega_j|\mathbf{x}) \tag{3.12}$$

Para este tipo de función de pérdida, la regla de decisión de Bayes puede simplificarse de la siguiente forma[1]: asignar un patrón de entrada $\mathbf{x}$ a la clase ω_j si

$$P(\omega_j|\mathbf{x}) > P(\omega_i|\mathbf{x}), \forall j \neq i \tag{3.13}$$

Si las probabilidades condicionadas de las clases son conocidas perfectamente, entonces, la regla de decisión bayesiana óptima puede utilizarse para construir el clasificador. Sin embargo, en la práctica, las probabilidades condicionadas son desconocidas y deben ser obtenidas de los patrones de entrenamiento disponibles. Si el modelo estadístico de las probabilidades es conocido (p.ej. distribuciones normales mutivariantes), pero algunos de los parámetros (p.ej. la media y la matriz de covarianza) son desconocidos, entonces, se trata, como se ha dicho anteriormente, de un problema de decisión paramétrico (clasificador supervisado paramétrico). Una estrategia común en este tipo de problemas es sustituir los parámetros desconocidos por sus valores estimados.

Cuando no se conoce el modelo estadístico de los datos, esto es, no puede suponerse un modelo paramétrico para las funciones de densidad de probabilidad asociadas a las clases, se utilizan modelos no paramétricos para estimar las funciones de densidad de probabilidad. En estos casos la única información disponible para la clasificación es el conjunto de prototipos. Un método no paramétrico muy extendido son los estimadores de Parzen. Otro método muy utilizado, tal vez el más simple entre los no paramétricos, es la regla k-NN o k-vecinos más cercanos [174], que consiste en etiquetar un patrón con la etiqueta del prototipo más cercano. De hecho, el perceptrón multicapa puede considerarse también como un método supervisado no paramétrico que construye regiones de decisión entre las clases [23].

Las diversas dicotomías que surgen en el campo del reconocimiento de patrones estadístico se muestran en la figura 3.3 mediante una estructura en forma de árbol. Si el gráfico se recorre de arriba a abajo y de izquierda a derecha, existe menos información disponible para la construcción del clasificador y por tanto, el problema se complica.

3.1.3.2. Clasificadores supervisados

El enfoque más simple e intuitivo para el diseño de un clasificador es el basado en el concepto de similitud: patrones que son similares deben asignarse a la misma clase. Por lo tanto, una vez que se ha establecido una buena métrica para definir esta similitud, los

[1]La regla definida en la ecuación 3.13 se conoce como regla MAP (*maximum a posteriori*)

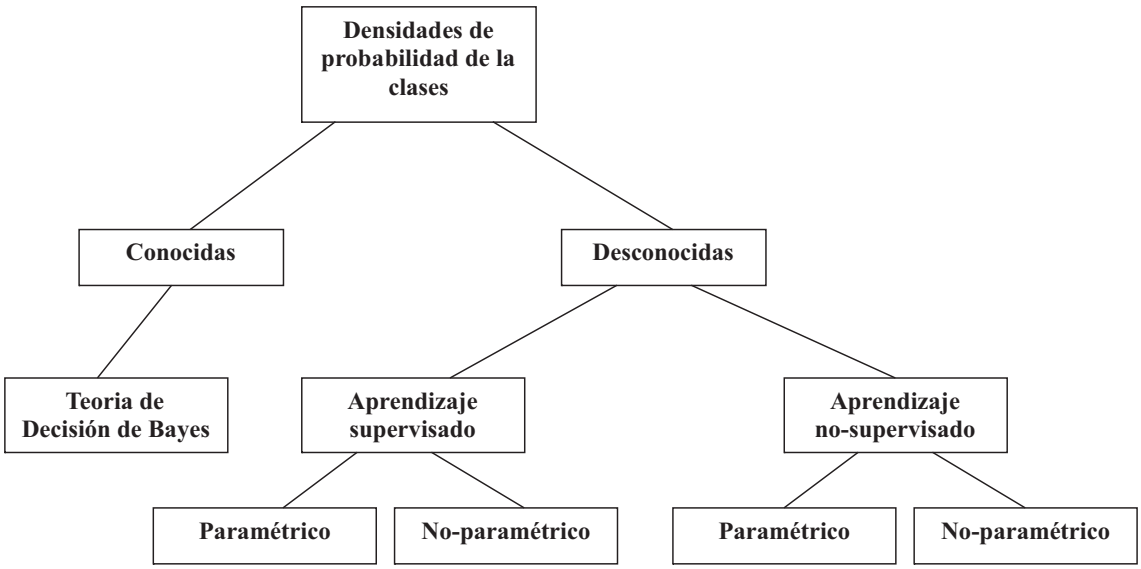


Figura 3.3: Diversos enfoques en el reconocimiento estadístico de patrones.

patrones pueden clasificarse mediante una correspondencia de modelos (*template matching*) o utilizando un clasificador de mínima distancia que utiliza un cierto número de prototipos por cada clase. La elección de la métrica y de los patrones de entrenamiento es crucial en el éxito de este enfoque de clasificación. En el clasificador basado en la media más cercana, la elección de los prototipos es muy simple y robusta; cada clase es representada por un prototipo individual el cual es el vector media de todos los patrones de esa clase. Entre técnicas más avanzadas en el cálculo de prototipos destacan el método VQ (*Vector Quantization*) [123], [176] y el LVQ (*Learning Vector Quantization*) [92], y los métodos de reducción asociados con la regla del vecino más cercano (1-NN), tales como la edición y condensación [42].

Un tipo especial de clasificador es el árbol de decisión [36] [138]. Se trata de un método de aprendizaje supervisado no paramétrico que destaca por su sencillez. Un árbol de decisión es una forma de representar el conocimiento obtenido en el proceso de aprendizaje inductivo. Puede verse como la estructura resultante de la partición recursiva del espacio de representación a partir del conjunto (numeroso) de prototipos. Esta partición recursiva se traduce en una organización jerárquica del espacio de representación que puede modelarse mediante una estructura tipo árbol. Cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada nodo hoja se refiere a una decisión (clasificación). Los árboles más comunes son binarios y consecuentemente, son intrínsecamente subóptimos para la mayoría de las aplicaciones. Sin embargo, la mayor ventaja del árbol de decisión, además de su velocidad, es la posibilidad de interpretar las reglas de decisión en términos de los atributos individuales. Los árboles de decisión más extendidos en su uso son CART (*Classification And Regression*

Trees)[27] y C4.5 [138].

3.1.3.3. Clasificadores no supervisados

Los clasificadores no supervisados surgen en situaciones donde el objetivo es construir regiones de decisión basadas en datos de entrenamiento no etiquetados, esto es, las clases de los patrones de entrenamiento son desconocidas a priori. La clasificación no supervisada también se conoce como búsqueda de agrupamientos, y en general, designa a una amplia variedad de procedimientos diseñados para encontrar agrupamientos naturales (*clusters*) en los datos multidimensionales, y que están basados en medidas de similitud entre los patrones [84].

La clasificación no supervisada es una tarea realmente compleja debido a que los datos pueden revelar agrupamientos de diferentes formas y tamaños. Además el número de agrupamientos en los datos suele depender de la resolución con la que se considera la información. Existen diversas definiciones funcionales de agrupamientos entre las que destacan las dos siguientes:

- Los patrones dentro de un agrupamiento son más similares entre ellos que los patrones pertenecientes a agrupamientos diferentes.
- Un agrupamiento consiste en una densidad de puntos relativamente elevada separada de otros agrupamientos por una densidad de puntos relativamente baja.

A pesar de la simplicidad de las definiciones funcionales anteriores, no es tan sencillo obtener una definición operacional para los agrupamientos. La mayor dificultad de estas técnicas reside en seleccionar una medida de similitud apropiada para definir a un agrupamiento y que tenga en cuenta los propios datos que conforman el agrupamiento y el contexto exterior de información.

Existen cientos de algoritmos no supervisados, sin embargo, la mayoría de ellos están basados en una de las dos técnicas de agrupamiento más populares: métodos de agrupamiento basados en el criterio SSE (*Sum of Squared Errors*) o suma de los errores al cuadrado; y agrupamientos basados en grafos (también conocidos como métodos de agrupamiento jerárquicos). En los primeros, el criterio SSE intenta obtener particiones de los datos en las que se minimice la varianza dentro de un agrupamiento y se maximice la varianza entre agrupamientos distintos. Para garantizar una solución óptima, se han de examinar todas las posibles particiones de los patrones en los K agrupamientos (para un K dado), lo que no siempre es posible debido a problemas computacionales. Así, en la práctica, se utilizan métodos heurísticos para reducir la búsqueda original, aún a costa de no garantizarse la optimalidad del procedimiento. En las técnicas jerárquicas los datos se organizan en secuencias de grupos anidados que pueden visualizarse en la forma de árboles. En la tabla 3.3 se muestra un listado de algunos de los métodos de agrupamiento más populares [84].

Métodos SSE	**Características**	**Comentarios**
K-medias	Genera agrupamientos hiperesféricos. Computacionalmente eficiente.	Necesita establecer el valor de K y los centros iniciales de los agrupamientos
Agrupamiento secuencial	Similar a las K-medias. Parte de un único agrupamiento y crea otros en base a una distancia.	Se considera un valor máximo para K
Algoritmo ISODATA	Basado en las K-medias, incorpora 3 funciones heurísticas: elimina, mezcla y divide agrupamientos	Requiere parámetros de ajuste de las funciones heurísticas
Grafos	**Características**	**Comentarios**
Basado en matriz de similaridad	Matriz cuadrada que expresa el grado de similaridad entre parejas de patrones	El espacio de almacenamiento es muy elevado
Cálculo del árbol generador minimal (AGM)	Basado en una medida de distancia entre agrupamientos	El agrupamiento se caracteriza por un miembro, y no por su valor medio

Tabla 3.3: Clasificadores no supervisados: principales métodos de agrupamiento: agrupamientos basados en el criterio SSE y basados en grafo.

3.1.4. Redes neuronales

Las redes neurales proporcionan toda una colección de algoritmos no lineales útiles para la extracción de características (mediante el uso de capas ocultas) y la clasificación de patrones. Algunos de los algoritmos de clasificación y extracción de características pueden ser mapeados en arquitecturas de redes neuronales para una implementación más eficiente. A pesar, aparentemente, de los diferentes principios subyacentes en cada uno de los enfoques, la mayoría de los modelos de redes neuronales son implícitamente equivalentes o similares a algunos de los métodos estadísticos clásicos[2]. A modo de ejemplo, la tabla 3.4 muestra la conexión entre algunos métodos estadísticos y los modelos de redes neuronales. En [6], Anderson afirma que *las redes neuronales son métodos estadísticos para aficionados... la mayoría de las RR.NN. ocultan la estadística al usuario* .

Métodos Estadísticos	**RR.NN.**
Funciones lineales discriminantes	Perceptrón
PCA	Redes autoasociativas
Estimación de la probabilidad a posteriori	Perceptrón Multicapa
Análisis discriminante no lineal	Perceptrón Multicapa
Regla K-NN	LVQ
Clasificadores basados estimadores de Parzen	RBF

Tabla 3.4: Conexiones entre los métodos estadísticos clásicos y los métodos basados en redes neuronales

[2]Sólo en cuanto al entrenamiento basado en descenso de gradiente. Una vez fijada la estructura de una red neuronal, encontrar los pesos óptimos por descenso del gradiente equivale a encontrar los pesos de máxima verosimilitud [112]

3.2. Análisis de Componentes Principales, PCA

El Análisis de Componentes Principales (PCA) junto con la relacionada Transformada Karhunen-Loève, o de Hotelling, o la factorización SVD, son técnicas clásicas muy utilizadas en el análisis estadístico de datos, la extracción de características y la compresión de datos. Los orígenes del PCA se sitúan en el trabajo de Pearson [135] y su primera aplicación es descrita en un artículo de Spearman [150]. Fue formulado con detalle por Hotelling [74] en 1933 y desde entonces es utilizado en numerosas aplicaciones. En el campo del reconocimiento de patrones, cabe destacar el éxito del PCA en el reconocimiento e identificación de caras humanas (*mugshots*) a partir de los trabajos de Turk y Pentland en [157] y Kirby y Sirovich en [91].

El análisis mediante componentes principales es una transformación lineal desde un espacio origen de dimensión muy elevada a un espacio de dimensiones más reducidas. El sistema de referencia del espacio original es rotado de manera de que la diversidad e información de los datos originales puede ser descrita eficientemente utilizando sólo un subconjunto de vectores.

Si los datos están muy correlados en el espacio original, PCA produce muy buenos resultados. Una alta correlación de los vectores de entrada significa que existe información redundante. PCA disminuye la redundacia mediante la incorrelación de los vectores de entrada: las nuevas coordenadas de los vectores originales en el nuevo espacio, esto es, las componentes principales, no están correladas. Por lo tanto, los vectores de entrada de dimensión elevada y correlados pueden ser eficientemente representados en un espacio de dimensiones inferior donde los vectores no están correlados. De esta manera, el PCA es una poderosa herramienta para la compresión de datos.

Este apartado contiene una descripción detallada del análisis de componentes principales utilizando dos planteamientos diferentes del problema: por maximización de la varianza y por minimización del error cuadrático medio. Para más información, se pueden consultar [63] y [72].

3.2.1. PCA por maximización de la varianza

Supongamos que se dispone de N vectores $\mathbf{x}_j$ de dimensión M alineados en una matriz de datos $\mathbf{X} \in \Re^{M \times N}$. Sea $\mathbf{u}$ una dirección de $\Re^M$. La proyección del vector $\mathbf{x}_j$ sobre el vector $\mathbf{u}$ puede calcularse de la siguiente forma:

$$p_j =< \mathbf{x}_j, \mathbf{u} >= \mathbf{u}^T \mathbf{x}_j = \sum_{i=1}^{M} u_i x_{ij} \tag{3.14}$$

El objetivo es encontrar la dirección $\mathbf{u}$ que maximiza la varianza de las proyecciones de todos los vectores de entrada $\mathbf{x}_j$, $j = 1, \ldots N.$, por lo tanto la función a maximizar es:

$$J^{PCA}(\mathbf{u}) = \sigma^2(p_j) = E\{p_j^2\} \tag{3.15}$$

Si la media de las proyecciones se calcula según el siguiente estimador:

$$\overline{p} = \frac{1}{N}\sum_{j=1}^{N} p_j = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M} u_i x_{ij} = \sum_{i=1}^{M} u_i \mu_i \tag{3.16}$$

la varianza a maximizar es igual a [3]:

$$\begin{aligned} J^{PCA}(\mathbf{u}) &= \sigma^2(p_j) = E\{p_j^2\} = \\ &= \frac{1}{N}\sum_{j=1}^{N}(p_j - \overline{p})^2 = \frac{1}{N}\sum_{j=1}^{N}\Big(\sum_{i=1}^{M} u_i x_{ij} - \sum_{i=1}^{M} u_i \mu_i\Big)^2 = \\ &= \frac{1}{N}\sum_{j=1}^{N}\Big(\sum_{i=1}^{M} u_i \hat{x}_{ij}\Big)^2 = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{l=1}^{M} u_i \hat{x}_{ij} u_l \hat{x}_{lj} = \\ &= \sum_{i=1}^{M}\sum_{l=1}^{M} u_i u_l \frac{1}{N} < \hat{\mathbf{x}}_{i'}, \hat{\mathbf{x}}_{l'} > = \sum_{i=1}^{M}\sum_{l=1}^{M} u_i c_{il} u_l = \mathbf{u}^T \mathbf{C}_{\mathbf{x}} \mathbf{u} \end{aligned} \tag{3.17}$$

En la ecuación anterior, μ_i es la media de la fila i de la matriz de datos $\mathbf{X}$ y $\hat{x}_{ij}$ es el valor de x_{ij} sin el valor de la media μ_i. Así el vector μ está formado por las medias de todas las filas:

$$\mu = [\mu_1, \ldots, \mu_M]^T = \frac{1}{N}\sum_{j=1}^{N} x_j \tag{3.18}$$

por lo tanto [4]:

$$\hat{\mathbf{X}} = \mathbf{X} - \mu \mathbf{1}_{1\times N} \tag{3.19}$$

y $\mathbf{C}_{\mathbf{x}}$ es la matriz de covarianzas de $\mathbf{X}$:

$$\mathbf{C}_{\mathbf{x}} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T \tag{3.20}$$

Como ya se ha dicho, el objetivo es maximizar la función 3.17 y además bajo la restricción que $\|\mathbf{u}\| = 1$. Puesto que la varianza depende tanto de la norma como de la orientación del vector $\mathbf{u}$, y aumenta sin límites si la norma también aumenta, cabe imponer la restricción de que la norma de $\mathbf{u}$ sea constante y en la práctica, se elige igual a la unidad. Si se utiliza la técnica de los multiplicadores de Lagrange, la función a maximizar es la siguiente:

[3] $\mathbf{x}_i$ es el vector columna i de la matriz $\mathbf{X}$, mientras que $\mathbf{x}_i^T$ denota al vector fila i de la matriz $\mathbf{X}$

[4] $\mathbf{1}_{M\times N}$ denota una matriz de dimension $M \times N$ donde todos los elementos tienen valor unidad

$$F(\mathbf{u},\lambda) = \mathbf{u}^T\mathbf{C_x}\mathbf{u} - \lambda(\mathbf{u}^T\mathbf{u}-1) = \sum_{i=1}^{M}\sum_{j=1}^{M} u_i c_{ij} u_j - \lambda\Big(\sum_{i=1}^{M} u_i^2 - 1\Big) \tag{3.21}$$

Una posible solución a este problema de maximización puede ser la siguiente:

$$\begin{aligned}\frac{\partial F}{\partial u_l} = \sum_{j=1}^{M} c_{lj}u_j + \sum_{i=1}^{M} u_i c_{il} - \lambda 2u_l = 0; \qquad & l = 1\ldots M \\ \sum_{i=1}^{M} c_{li}u_i = \lambda u_l; \qquad & l = 1\ldots M \\ \mathbf{C_x}\mathbf{u} = \lambda\mathbf{u} & \end{aligned} \tag{3.22}$$

Por lo tanto, para encontrar los valores de $\mathbf{u}$ y λ que maximizan 3.21 se deben calcular los vectores propios y los valores propios de la matriz de covarianzas de los datos $\mathbf{C_x}$. El valor propio mayor es igual a la máxima varianza, mientras que su correspodiente vector propio determina la dirección de la máxima varianza en los datos.

Para diagonalizar la matriz de covarianzas de los datos $\mathbf{C_x}$ se realiza la descomposición en valores singulares (SVD) de la matriz $\mathbf{C_x}$:

$$\mathbf{C_x} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{3.23}$$

en la expresión anterior la matriz ortonormal $\mathbf{U} = [\mathbf{u_1},\ldots,\mathbf{u_R}] \in \Re^{M\times R}$ contiene en sus columnas a los vectores propios $\mathbf{u_1},\ldots,\mathbf{u_R}$ y la matriz diagonal $\mathbf{\Lambda} \in \Re^{R\times R}$ contiene en la diagonal a los valores propios $\lambda_1,\ldots,\lambda_R$, siendo R el rango de la matriz $\mathbf{C_x}$.A efectos prácticos, los valores propios y sus correspondientes vectores propios se ordenan de mayor a menor según el valor de los valores propios, es decir, $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_R$. Con este orden, la mayor variabilidad en los datos está reflejada en los primeros vectores propios. Por lo tanto, los vectores propios se suelen también denominar *vectores principales* (o también *ejes principales* o *direcciones principales*).

Esta forma de calcular los vectores principales es la más clara y la más utilizada. Sin embargo, si el tamaño del vector de datos, M, es muy grande, que es el caso de las imágenes en visión por computador, la matriz de covarianzas $\mathbf{C_x} \in \Re^{M\times M}$ es demasiado grande y su diagonalización no es factible computacionalmente. Pero existe una solución: si el número de vectores de datos es menor que el tamaño de estos vectores ($N < M$), puede realizarse el análisis PCA de la matriz de datos original utilizando el método propuesto por Murakami y Kumar [117], y conocido en la literatura por *truco de Turk y Pentland*

[157] y que consiste en un simple truco algebraico. En lugar de utilizar la matriz de covarianzas de los datos $\mathbf{C_x} \in \Re^{M \times M}$ (producto externo de la matriz de datos) se utiliza la matriz de covarianzas $\mathbf{C_x}' \in \Re^{N \times N}$ (producto interno de la matriz de datos):

$$\mathbf{C_x}' = \frac{1}{N}\hat{\mathbf{X}}^T\hat{\mathbf{X}} \tag{3.24}$$

Los vectores propios y valores propios de la matriz de covarianzas $\mathbf{C_x}$ pueden calcularse a partir de los vectores propios $\mathbf{u}'_i$ y valores propios λ'_i de $\mathbf{C_x}'$ de la siguiente manera:

$$\lambda_i = \lambda'_i; \quad i = 1 \ldots N \tag{3.25}$$

$$\mathbf{u}_i = \frac{\hat{\mathbf{X}}\mathbf{u}'_i}{\sqrt{N\lambda_i}}; \quad i = 1 \ldots N \tag{3.26}$$

La matriz $\mathbf{C_x}'$ es mucho menor que la original $\mathbf{C_x}$ cuando $N << M$. Por lo tanto, la descomposición en vectores y valores propios de la matrix original $\mathbf{C_x} \in \Re^{M \times M}$ se ha reducido a la descomposición más factible de $\mathbf{C_x}' \in \Re^{N \times N}$.

3.2.2. Propiedades de PCA

La matriz ortonormal $\mathbf{U}$ que contiene los vectores principales puede utilizarse como una transformación lineal para proyectar los datos originales de un espacio de entrada de muy alta dimensionalidad en otro espacio de más baja dimensionalidad (*subespacio*) y viceversa. Las columnas de $\mathbf{U}$ son la base de vectores del nuevo subespacio de menor dimensión expresadas en coordenadas multidimensionales del espacio original. De esta manera, un vector de datos puede ser proyectado en el subespacio principal utilizando la matriz de transformación $\mathbf{U}^T : \Re^M \rightarrow \Re^N$.

$$\mathbf{p} = \mathbf{U}^T \hat{\mathbf{x}} \tag{3.27}$$

Los coeficientes p_j se calculan como las proyecciones de imagen de entrada sobre el vector principal:

$$p_j =< \hat{\mathbf{x}}_j, \mathbf{u}_j >= \mathbf{u}_j^T \hat{\mathbf{x}}_j = \sum_{i=1}^{M} u_{ij} \hat{x}_i; \quad j = 1 \ldots N \tag{3.28}$$

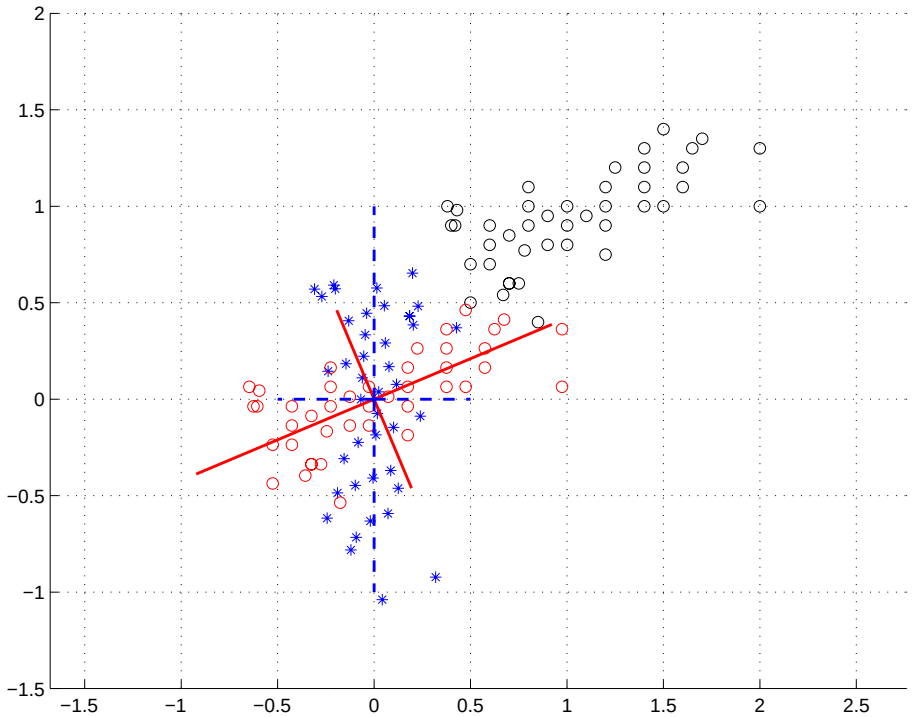


Figura 3.4: Ejemplo ilustrativo de la transformación PCA: datos originales (puntos negros), datos centrados (puntos rojos), datos proyectados en las direcciones de los vectores propios (puntos azules), los ejes principales están en azul.

Todos los vectores de entrada contenidos en la matriz de datos centrada $\hat{\mathbf{X}}$ pueden ser proyectados,

$$\mathbf{P} = \mathbf{U}^T \hat{\mathbf{X}} \tag{3.29}$$

y la matriz que contiene las proyecciones $\mathbf{P}$ es una transformación ortonormal de los datos centrados $\hat{\mathbf{X}}$, así que la media de las proyecciones también está centrada:

$$\mu_{\mathbf{P}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{p}_j = \frac{1}{N}\sum_{j=1}^{N} \mathbf{U}^T \hat{\mathbf{x}} = \mathbf{U}^T \frac{1}{N}\sum_{j=1}^{N} \hat{\mathbf{x}} = 0 \tag{3.30}$$

La figura 3.4 ilustra el proceso de la transformación PCA de unos datos artificiales: la información original son los puntos negros, los datos centrados se representan en rojo, los ejes principales (los vectores propios que definen la transformación) se muestran en azul y los datos proyectados en el nuevo espacio son los puntos azules.

Ahora, calculemos la matriz de correlación de las proyecciones:

$$\begin{aligned} \mathbf{C}_{\mathbf{P}} &= \frac{1}{N}\mathbf{P}\mathbf{P}^T = \frac{1}{N}\mathbf{U}^T\hat{\mathbf{X}}(\mathbf{U}^T\hat{\mathbf{X}})^T = \mathbf{U}^T\frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T\mathbf{U} = \\ &= \mathbf{U}^T\mathbf{C}_{\mathbf{x}}\mathbf{U} = \mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{U} = \mathbf{\Lambda} \end{aligned} \tag{3.31}$$

En el desarrollo anterior se ha sustituido la matriz $\mathbf{C}_{\mathbf{x}}$ por su forma diagonal (3.23) y se ha considerado la ortonormalidad de la matriz $\mathbf{U}$ (es decir que, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$). Por lo tanto, la matriz de covarianzas de los datos transformados (o proyecciones) es igual a la matriz diagonal $\mathbf{\Lambda}$, que contiene a los valores propios en su diagonal. Este hecho tiene dos consecuencias importantes, la primera de ellas, que demuestra que los vectores transformados están no correlados. Luego la redundancia debida a la correlación entre los vectores de entrada se ha eliminado. Y la segunda consecuencia es que se demuestra que la varianza en la dirección del eje principal i (la varianza de cada una de las i componentes principales) es igual al valor del valor propio i, es decir, $\frac{1}{N}\sum_{j=1}^{N} p_{ij}^2 = \lambda_i$.

Una propiedad importante de la diagonalización (3.23) es que se preserva la traza de la matriz que ha sido diagonalizada. Dado que la suma de los elementos de la diagonal de la matriz de covarianzas es igual a la suma de las varianzas de los vectores de entrada, esto implica que la varianza total de los datos de entrada se ha conservado en la transformación y es igual a la suma de todos los valores propios:

$$VAR(\mathbf{X}) = \sum_{i=1}^{M} \frac{1}{N} \hat{\mathbf{x}}_i \hat{\mathbf{x}}_i^T = \sum_{i=1}^{M} c_{ii} =$$
$$= \sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \frac{1}{N} \mathbf{p}_{i'} \mathbf{p}_{i'}^T = VAR(\mathbf{P}) \tag{3.32}$$

A continuación se explica cómo utilizar la matriz de vectores propios $\mathbf{U}$ para proyectar un conjunto de coeficientes del subespacio en el espacio original; a esta operación se la denomina *reconstrucción*. El vector de coeficientes o proyecciones $\mathbf{p}$ puede ser reconstruido utilizando la matriz de transformación $\mathbf{U} : \Re^N \rightarrow \Re^M$:

$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{p} = \sum_{j=1}^{N} p_j \mathbf{u}_j \tag{3.33}$$

Si se utilizan los N vectores propios de $\mathbf{U} \in \Re^{M \times N}$ para la reconstrucción, cada imagen de $\mathbf{X}$ puede ser perfectamente reconstruida sin ningún tipo de error en la reconstrucción. Pero lo que es más interesante es el hecho que una imagen pueda ser también reconstruida utilizando sólo un subconjunto de las componentes principales. Esto es posible debido a la distribución de la varianza en los ejes principales: la mayor parte de la varianza está contenida en los primeros vectores propios. Este hecho puede representarse utilizando una fracción de la varianza total del conjunto de datos que se suele denominar como la *energía* contenida en los primeros k vectores propios:

$$energia_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{3.34}$$

Por lo tanto, es posible obtener una buena aproximación de las imágenes de entrada considerando solo un subconjunto de vectores propios (los asociados a los eigevalores de mayor magnitud). Luego, si se consideran sólo k vectores propios o componentes principales, $k < N$, entonces el subespacio a utilizar se reduce a $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \Re^{M \times k}$. Un vector de entrada puede ser proyectado en este subespacio de componentes principales de dimensión k utilizando la matriz de transformación $\mathbf{U}^T : \Re^M \rightarrow \Re^k$:

$$\begin{aligned} \mathbf{p} &= \mathbf{U}^T \hat{\mathbf{x}} = \mathbf{U}^T (\mathbf{x} - \mu) \\ p_j &= < \hat{\mathbf{x}}_j, \mathbf{u}_j > = \mathbf{u}_j^T \hat{\mathbf{x}}_j = \sum_{i=1}^{M} u_{ij} \hat{x_i} = \sum_{i=1}^{M} u_{ij} (x_i - \mu_i); \quad j = 1 \ldots k \end{aligned} \tag{3.35}$$

y reconstruirse utilizando la transformación $\mathbf{U} : \Re^k \rightarrow \Re^M$:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{U}\mathbf{p} = \sum_{j=1}^{k} p_j \mathbf{u}_j \\ \mathbf{y} &= \hat{\mathbf{y}} + \mu \end{aligned} \tag{3.36}$$

De esta manera un imagen puede aproximarse por una combinación lineal de los primeros k vectores propios.

La figura 3.5 ilustra el proceso completo de PCA descrito anteriormente: generación del subespacio y reconstrucción. En la parte superior, la primera fila recoge las imágenes en color de nueve conocidos pilotos de motociclismo [5], son las imagenes de entrada utilizadas para generar el subespacio PCA, es decir forman la matriz de datos $\mathbf{X} \in \Re^{M \times N}$, donde $N = 9$, es el número de pilotos y $M = 273 \times 254 \times 3 = 208026$, es la dimensión de cada una de ellas. En la segunda fila, se muestran los ocho vectores propios obtenidos a partir del conjunto de imágenes anterior, $\mathbf{U}$, y la media[6] de éste, μ. En el resto de filas se muestran la reconstrucciones, $\mathbf{y}$, de cada una de las imágenes originales a partir de subconjuntos del eigenespacio, es decir, se muestra la reconstrucción de cada imagen utilizando subconjuntos de vectores propios compuestos por $1, 2, \ldots, 8$ vectores, la imagen situada a la izquierda es la reconstruida a partir de un sólo coeficiente y un sólo vector propio, y la situada más a la derecha es la recontruida a partir de todo el subespacio y por tanto es igual a la original. Al final de cada conjunto de imágenes reconstruidas se muestra el error de reconstrucción de cada una de ellas (se define en 3.38).

El vector de error de reconstrucción es igual a la diferencia entre el vector de entrada original y el reconstruido :

$$\mathbf{e} = \hat{\mathbf{x}} - \hat{\mathbf{y}} = \sum_{j=1}^{N} p_j \mathbf{u}_j - \sum_{j=1}^{k} p_j \mathbf{u}_j = \sum_{j=k+1}^{N} p_j \mathbf{u}_j \tag{3.37}$$

La medida de error más común es el cuadrado del vector del error de reconstrucción, el cual es igual a su norma L2 al cuadrado; si consideramos la ortonormalidad de los vectores propios $\mathbf{u}_j$ se obtiene:

$$e = \|\mathbf{e}\|^2 = \| \sum_{j=k+1}^{N} p_j \mathbf{u}_j \|^2 = \sum_{j=k+1}^{N} p_j^2 \tag{3.38}$$

[5]Los pilotos de las imágenes son de izdqa a dra: Valentino Rossi, Sete Gibernau, Dani Pedrosa, Jorge Lorenzo, Loris Capirossi, Max Biaggi, Toni Elías, Roberto Rolfo y Manuel Poggiali

[6]En la segunda fila, la imagen media es la última imagen de la derecha.

Por lo tanto, el error de reconstrucción es igual a la suma de los cuadrados de las coeficientes rechazados. En el ejemplo anterior, (figura 3.5), se muestra el error de cada una de las reconstrucciones de cada uno de los pilotos, y se aprecia como el error es diferente para cada uno de ellos, así para el primer piloto (Rossi), el error de reconstrucción es muy elevado en todas las reconstrucciones y sólo se elimina al utilizar el espacio de vectores propios completo, mientras que para otros pilotos, como el quinto (Capirossi), se obtiene una buena reconstrucción con solo 4 vectores propios.

Normalmente, los coeficentes rechazados no son conocidos, puesto que se desprecian sus vectores propios asociados y los coeficientes no se calculan, sin embargo, es posible estimar el valor del error de reconstrucción con los valores de las varianzas a lo largo de los vectores propios rechazados, que son iguales a sus valores propios correspondientes:

$$E\{e\} = \sum_{j=k+1}^{N} \lambda_j \tag{3.39}$$

De esta forma, la esperanza del error es igual a la suma de los valores propios despreciados en la reconstrucción. Esta última afirmación confirma el hecho que maximizando la varianza de los primeros vectores propios (los que se utilizan para la reconstrucción), el error de reconstrucción está siendo minimizado. Esta consideración da lugar a la posibilidad de plantear el problema de PCA desde el punto de vista de la minimización del error cuadrático medio, que se discutirá en el siguiente apartado.

3.2.3. PCA por minimización del error cuadrático medio

Como alternativa a la maximización de la varianza, los vectores principales pueden estimarse a partir de la minimización del error cuadrático medio:

$$J^{PCA}(\mathbf{u}) = E\{\|\mathbf{e}\|^2\} = E\{\|\hat{\mathbf{x}} - \hat{\mathbf{y}}\|^2\} = E\{\|\hat{\mathbf{x}} - \sum_{j=1}^{N}(\mathbf{u}_j^T\hat{\mathbf{x}})\mathbf{u}_j\|^2\} \tag{3.40}$$

debido a la ortonormalidad de los vectores $\mathbf{u}_j$, la función anterior puede escribirse como:

$$\begin{aligned} J^{PCA}(\mathbf{u}) &= E\{\|\hat{\mathbf{x}}\|^2\} - E\{\sum_{j=1}^{N}(\mathbf{u}_j^T\hat{\mathbf{x}})^2\} = \\ &= trace(\mathbf{C_x}) - \sum_{j=1}^{N}\mathbf{u}_j^T\mathbf{C_x}\mathbf{u}_j \end{aligned} \tag{3.41}$$

En [43] se demuestra que un mínimo de (3.41), bajo la condición de ortonomalidad de los vectores $\mathbf{u}_j$, viene dado la base ortonormal compuesta por los N primeros vectores propios de (3.23). Sin embargo, el criterio de minimización de (3.41) no especifica que la anterior solución sea la única y de hecho cualquier base ortonormal del subespacio de los vectores propios de (3.23) proporciona la misma propiedad de máxima compresión en relación al error cuadrático medio. Aunque esta ambigüedad puede verse como una desventaja, existen otro tipo de criterios en los que un subespacio de PCA es preferible sobre otro, como por ejemplo, el criterio de independencia estadística impuesto por ICA [79] .

Figura 3.5: Esta figura ilustra el proceso de PCA. Arriba, la primera fila recoge las imágenes en color de 9 conocidos pilotos de motociclismo, y son las imagenes de entrada utilizadas para generar el eigespacio. En la segunda fila, se muestran los ocho vectores propios obtenidos y la imagen media. En el resto de filas se muestran las reconstrucciones junto con su error de reconstrucción.

3.2.4. Algunas consideraciones sobre PCA

PCA es una técnica de extracción de características poderosa y sencilla pero con una serie de limitaciones debidas a su propia naturaleza. La transformación PCA no puede distinguir conjuntos de puntos cuyas estadísticas de segundo orden sean iguales. Esta limitación se ilustra en la figura 3.6(a), donde se muestran cuatro conjuntos de datos diferentes cuyas componentes principales son iguales.

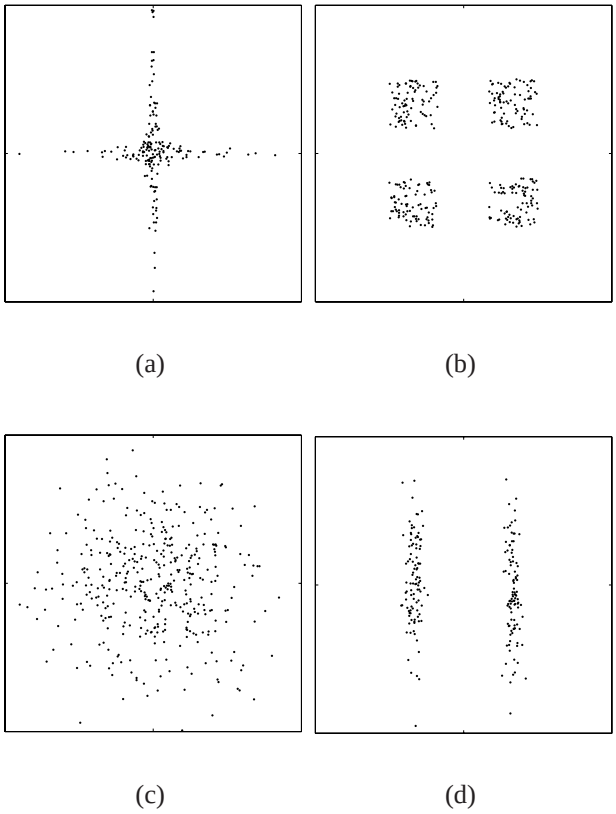


Figura 3.6: Ejemplos de conjuntos de datos diferentes cuyas componentes principales son iguales.

Además, puesto que se trata de una técnica no supervisada, esto es, la transformación es independiente de las clases existentes en los datos originales, los ejes principales no indican a priori las direcciones que mejor separan las clases. Por ejemplo, si consideramos el conjunto de datos de la figura 3.6(d) suponiendo que cada nube de puntos es una clase diferente, y pretendemos clasificar los puntos reduciendo previamente los datos a una sola dimensión, la proyección de los datos sobre la primera componente principal (el eje vertical) produce la confusion de las dos clases, mientras que la proyección sobre la segunda componente principal (el eje horizontal) permite la separación completa entre las

dos clases. En la siguiente sección, se mostrará como la técnica de LDA permite reducir el conjunto de puntos de la figura 3.6(d) a una sola dimensión manteniendo la separación entre clases.

3.3. Análisis de Fisher, LDA

El análisis de Fisher ([56]), análisis canónico o análisis discriminante lineal (LDA) [7] se centra en la búsqueda de aquellos vectores que describen el subespacio que mejor discrimina entre clases, a diferencia del análisis PCA, donde el subespacio definido por los vectores es aquel que mejor describe el conjunto de los datos. Dicho de manera más formal, dado un conjunto de N vectores $\mathbf{x}_i^j$ de dimensión M alineados en una matriz de datos $\mathbf{X} \in \Re^{M \times N}$ de forma que se disponen de C clases y K ejemplos por cada clase:

$$\mathbf{X} = [\mathbf{X^1}|\mathbf{X^2}|\ldots|\mathbf{X^j}|\ldots|\mathbf{X^C}] \tag{3.42}$$

$$\begin{aligned} \mathbf{X^1} &= [\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_i^1, \ldots, \mathbf{x}_K^1] \\ \mathbf{X^j} &= [\mathbf{x}_1^j, \mathbf{x}_2^j, \ldots, \mathbf{x}_i^j, \ldots, \mathbf{x}_K^j] \\ \mathbf{X^C} &= [\mathbf{x}_1^C, \mathbf{x}_2^C, \ldots, \mathbf{x}_i^C, \ldots, \mathbf{x}_K^C] \end{aligned} \tag{3.43}$$

se desea encontrar una matriz de transformación $\mathbf{W} : \Re^M \rightarrow \Re^P$ que describa el mejor subespacio que discrimine entre clases, tras proyectar los datos originales en ese nuevo espacio definido por $\mathbf{W}^T$:

$$\mathbf{p} = \mathbf{W}^T \mathbf{x}_i^j \tag{3.44}$$

Dado un número de características independientes relativas a los datos, LDA crea una combinación de éstas en las que se incrementan las diferencias medias entre las clases establecidas. Para todos los ejemplos de todas las clases se definen dos matrices, la matriz de covarianza interna media [8] y la matriz de covarianza entre las clases [9].

La matriz de covarianza interna media viene dada por la siguiente expresión:

$$\mathbf{S_w} = \sum_{j=1}^{C} \sum_{i=1}^{K} (\mathbf{x}_i^j - \mu_j)(\mathbf{x}_i^j - \mu_j)^T = \sum_{j=1}^{C} \hat{\mathbf{X}}^{\mathbf{j}} \hat{\mathbf{X}}^{\mathbf{j}^T} \tag{3.45}$$

donde

$$\hat{\mathbf{X}}^{\mathbf{j}} = \mathbf{X^j} - \mu_{\mathbf{j}} \mathbf{1}_{1 \times N} \tag{3.46}$$

y μ_j es la media de la clase j.

La otra matriz, la matriz de covarianza entre las clases, se define como:

[7] Linear Discriminant Analysis
[8] "within-class"scatter matrix
[9] "between class"scatter matrix

$$\mathbf{S_b} = \sum_{j=1}^{C} (\mu_j - \mu)(\mu_j - \mu)^T \tag{3.47}$$

donde μ es la media de todos los ejemplos de todas las clases.

El objetivo es maximizar la distancia entre las clases ($\mathbf{S_b}$) y minimizar las distancias entre ejemplos pertenecientes a la misma clase ($\mathbf{S_w}$). Una forma de conseguirlo es maximizar el cociente de los determinantes de ambas matrices:

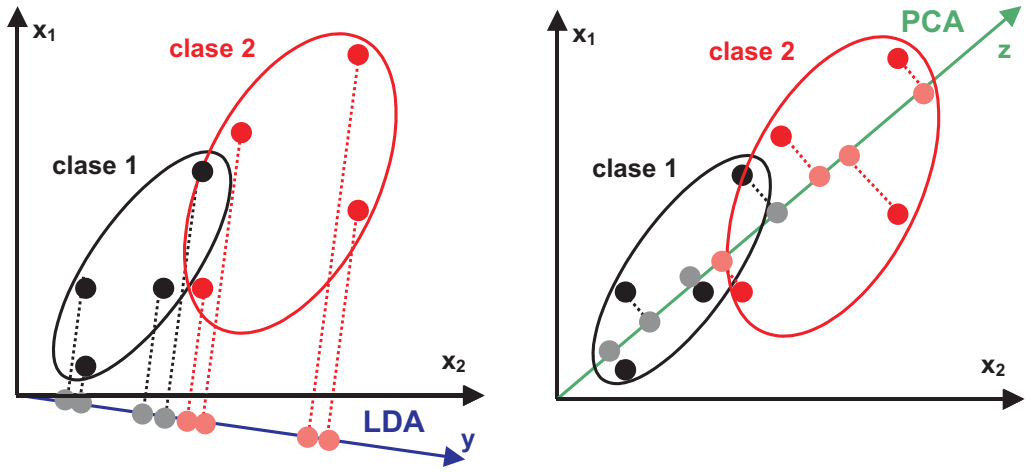
$$\mathbf{J^{LDA}} = \frac{det|\mathbf{S_b}|}{det|\mathbf{S_w}|} \tag{3.48}$$

La ventaja de utilizar la función 3.48 es que si la matriz de covarianza interna media ($\mathbf{S_w}$) es no singular entonces, el cociente anterior se maximiza cuando los vectores de la matriz de transformación $\mathbf{W}$ son los vectores propios de ${\mathbf{S_w}}^{-1}\mathbf{S_b}$ [56]. Por lo tanto, los vectores de la transformación se obtienen a partir de la descomposición en valores singulares de ${\mathbf{S_w}}^{-1}\mathbf{S_b}$:

$${\mathbf{S_w}}^{-1}\mathbf{S_b} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T \tag{3.49}$$

Se demuestra que como máximo existen $C-1$ vectores propios no nulos y por tanto un límite superior para la dimensión P del nuevo subespacio es $C-1$ y se requieren al menos $M+C$ ejemplos para garantizar que la matriz $\mathbf{S_w}$ sea no singular y por tanto, invertible. Esta última condición es casi imposible de cumplir en la realidad, por ejemplo, si utilizamos como datos de entrada vectores de imágenes, por ejemplo de tamaño $M = 128 \times 128 = 16384$ se necesitarían $16384 * C$ ejemplos de cada una de las clases a separar para asegurar que la matriz $\mathbf{S_w}$ sea invertible y obtener así los vectores del nuevo subespacio discriminante. Para resolver este problema, [14] y [155] proponen el uso de un subespacio intermedio. En ambos casos, el subespacio intermedio elegido es el dado por PCA. Por tanto, en el caso de que los datos de entrada sean de dimensión muy elevada, el espacio original de dimensión M se proyecta en un subespacio de menor dimensión Q utilizando PCA y finalmente, se proyecta sobre un subespacio de dimensión P utilizando LDA.

La figura 3.7(a) ilustra el efecto de la transformación LDA sobre un conjunto de datos compuesto por dos clases. En el gráfico se ha trazado en azul el primer eje (o vector) proporcionado por LDA y se han proyectado los puntos originales en la recta definida por la transformación. El eje y (primer eje proporcionado por LDA) permite separar las dos clases fácilmente. La figura 3.7(b) también muestra el resultado de reducir los puntos originales a una sola dimensión, pero en este caso, haciendo uso de la primera componente



(a) El eje y (primer eje proporcionado por LDA) permite separar las dos clases.

(b) El eje z (primera componente principal proporcionado por PCA) no consigue la separación total entre las dos clases.

Figura 3.7: Transformaciones LDA y PCA sobre un conjunto de puntos compuesto por dos clases. Se aprecia como al reducir los puntos originales a una sola dimensión, la transformación LDA consigue separar las dos clases, mientras que la primera componente principal no lo consigue totalmente.

principal proporcionada por la transformación PCA. En este caso, la separación de los ejemplos de las dos clases no se realiza completamente.

Como vimos en la sección anterior, la transformación por componentes principales se basa en la matriz de covarianza global por lo que no considera la estructura particular de las clases. Aún así, es frecuentemente empleada como un método de extracción de características porque son frecuentes las aplicaciones en las que las clases se distribuyen en la dirección de la máxima dispersión de los datos [157] [118] [124]. La transformación LDA busca maximizar la separación entre clases y si los datos de entrenamiento son una buena representación de los datos del sistema, LDA debe mejorar los resultados de reconocimiento de PCA; pero si los datos utilizados para generar el subespacio no son representativos del conjunto total, en estas circunstancias, los resultados de LDA no siempre son mejores que los de PCA como se demuestra en el trabajo de Martinez y Kak [111].

La figura 3.3 ilustra esta limitación natural del método de LDA: en el conjunto de puntos original se dispone de dos clases de datos; si se utilizan sólo dos ejemplos de entrenamiento por cada clase, se obtienen las direcciones de PCA y LDA que se dibujan en trazo más oscuro. D_{PCA} y D_{LDA} representan los umbrales de decisión que separan cada

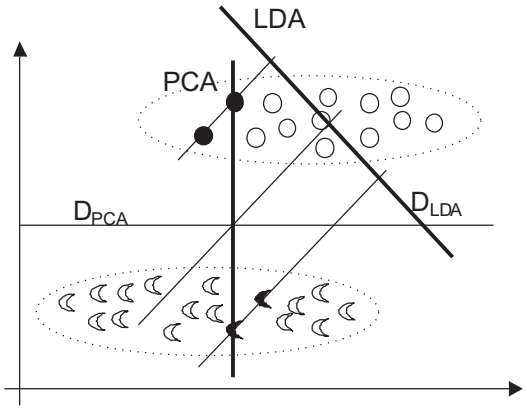


Figura 3.8: En este ejemplo, los resultados de clasificación de PCA (utilizando 1 solo vector propio) son superiores a los de LDA. [111]

clase. En el ejemplo, se aprecia como los resultados de clasificación de PCA (utilizando 1 solo vector propio) son superiores a los de LDA puesto que los ejemplos (entrenamiento) utilizados para la generación de los subespacios no son representativos de las clases. Esta circunstancia, el utilizar pocos ejemplos de entrenamiento en comparación con la dimensión de éstos, suele ser la común en las aplicaciones de reconocimiento basadas en la apariencia, por lo que en la práctica, LDA puede no mejorar significativamente los resultados de PCA.

Recientemente, en [35] , Cevikalp et al. proponen una variante de LDA que soluciona el problema de la singularidad de la matriz $\mathbf{S}_w$ en el caso de las imágenes, mediante el cálculo de vectores comunes discriminantes y el procedimiento de ortogonalización Gram-Schmidt. Con este nuevo algoritmo, se evita el cálculo de un subespacio intermedio por PCA, aunque, obviamente, no se salva la limitación señalada anteriormente.

3.4. Análisis de Componentes Independientes, ICA

ICA es una herramienta estadística de análisis multivariante relativamente nueva [87, 37] que intenta descubrir una cierta estructura latente en los datos. El modelo lineal de ICA ha sido estudiado de manera extensa por [37, 15, 33, 79]. En el método de PCA la redundancia se mide utilizando la correlación entre los datos, mientras que en ICA se utiliza el concepto más general de independencia estadística. Dicho de otra forma, la restricción estadística impuesta por PCA es la incorrelación de los datos, mientras que ICA impone la restricción de independencia estadística, la cual incluye la restricción de incorrelación impuesta por PCA y no a la inversa.

En este apartado se desarrolla ampliamente el concepto de análisis de componentes independientes y su aplicación a la extracción de características.

3.4.1. Modelo ICA

El modelo lineal de ICA es un modelo *generativo*, lo que significa que describe los datos originales como si estuviesen generados por un proceso de mezcla de componentes. Según este modelo, un vector de datos [10] $\mathbf{x}$ de dimensión M puede modelarse como:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{3.50}$$

donde $\mathbf{s}$ es el vector de variables estadísticamente independientes [11]:

$$p(\mathbf{s}) = \prod_{i=1}^{N} p_i(s_i) \tag{3.51}$$

$\mathbf{A}$ es una matriz de coeficientes constantes de tamaño $M \times N$ que se suele denominar *matriz mezcla* y $\mathbf{n}$ es un vector representativo de posible ruido aditivo independiente de $\mathbf{s}$. El anterior modelo se puede simplificar asumiendo que los datos originales no disponen de ruido, esto es:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.52}$$

3.4.2. Estimación del modelo ICA

Existen dos enfoques distintos respecto a cuál es el objetivo real que se logra al estimar las componentes independientes de un conjunto de datos multivariantes. En el primero de ellos se considera que los datos originales multivariantes están generados a partir de una combinación desconocida de unas señales fuente independientes, s_i, y la tarea consiste en

[10] $\mathbf{x}$ tiene media cero.

[11] El conjunto de los vectores $\mathbf{s}$ son las *fuentes* o *componentes independientes (IC's)*

su estimación a partir de los datos. Este punto de vista se suele denominar como separación ciega de fuentes[12]: existe una serie de fuentes que han sido mezcladas, y el proceso de mezcla es desconocido (de ahí el término *ciego*).

El otro enfoque considera que ICA es un método de representación de la información que permite presentar los datos en una forma más comprensible, revelando la estructura oculta de la información original y en algunos casos, posibilitando la reducción de su dimensión. Este otro punto de vista se puede denominar como enfoque de extracción de características o bien, considerar el problema como un caso de minería de datos.

En terminos generales, el objetivo de ICA es estimar las componentes independientes $\mathbf{s}$ y la matriz mezcla $\mathbf{A}$ utilizando para ello sólo la información del vector de datos $\mathbf{x}$. Si se utiliza el modelo de la ecuación 3.52, se puede apreciar que estimando sólo los elementos de la matriz mezcla es posible calcular las componentes independientes mediante la transformación:

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x} \tag{3.53}$$

Cabe considerar que es imposible determinar la varianza original de las componentes, su signo y su orden puesto que:

$$s_i := \alpha s_i \text{ es equivalente a } \mathbf{a}_i := \alpha \mathbf{a}_i$$

y

$$s_i \longleftrightarrow s_j \text{ es equivalente a } \mathbf{a}_i \longleftrightarrow \mathbf{a}_j$$

Se ha demostrado [37, 79] que las componentes independientes del modelo generativo de la ecuación 3.52 correspoden a las proyecciones unidimensionales no correladas que cumplen la condición de máxima no-gaussianidad. Dicho de otra forma, se desean encontrar vectores $\mathbf{w}_i$ tales que

$$s_i = \mathbf{w}_i^T \mathbf{x} \tag{3.54}$$

tengan distribuciones no-gaussianas máximas y esten mutuamente decorreladas, es decir:

$$E\{s_i s_j\} = 0, i \neq j \tag{3.55}$$

Una forma sencilla de calcular estos vectores $\mathbf{w}_i$ consiste en primero *blanquear* los datos, y luego buscar proyecciones ortogonales no-normales. Esto es correcto debido a que las proyecciones no correladas en el espacio original corresponden a proyecciones ortogonales de los datos blanqueados y viceversa. Por lo tanto es posible utilizar un proceso en dos pasos para estimar las componentes independientes:

[12]BSS, Blind source separation

1 El vector de datos $\mathbf{x}$ se transforma linealmente (proceso de blanqueo) en $\mathbf{y} = \mathbf{V}\mathbf{x}$, de tal forma que los elementos de $\mathbf{y}$ tienen varianza unidad y están decorrelados ($E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}$). Esta operación puede realizarse, por ejemplo [13], haciendo $\mathbf{V} = {\mathbf{C}_\mathbf{X}}^{-1/2}$ donde $\mathbf{C}_\mathbf{X} = E\{\mathbf{x}\mathbf{x}^T\}$ es la matriz de correlación de los datos.

2 Buscar una matriz ortogonal B que maximice alguna medida de no gaussianidad de los elementos $\mathbf{s} = \mathbf{B}\mathbf{y}$

3.4.3. Codificación *sparse*

El concepto de codificación *sparse* [55, 57] se refiere a un tipo de representación de la información donde sólo unas pocas componentes, pertenecientes a un vector mutidimensional de elevada dimensión, son utilizadas de manera efectiva para representar los datos. Esto implica que la mayoría de las componentes del vector están muy cerca del cero y sólo unas pocas tiene valores significativos. La figura 3.9 ilustra el concepto de *sparsity*.

Este tipo de codificación se desarrolló a partir de ciertas consideraciones neurofisiológicas [57, 156], y simulaciones de métodos basados en este tipo de codificación han proporcionado resultados que se corresponden adecuadamente con las medidas reales de las neuronas [127, 128].

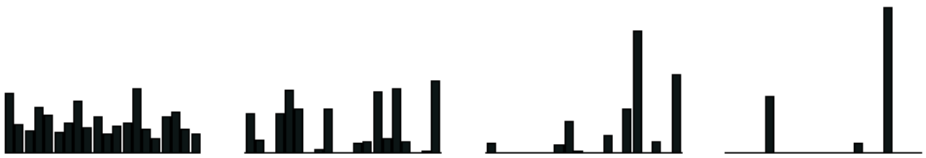


Figura 3.9: Se muestran cuatro ejemplos de vectores con distinto nivel de *sparsity*. Los niveles de *sparseness* que se muestran son 0.1, 0.4, 0.7 y 0.9. Cada barra denota el valor de cada uno de los elementos del vector. Si el nivel de *sparsity* es bajo, todos los elementos son igualmente activos, sin embargo, para valores de *sparsity* elevados, la mayoría de los coeficientes son cero y sólo unos pocos tienen valores significativos.

Olshausen y Field [127, 128] utilizaron un modelo generativo lineal (como el de la ecuación 3.52) seguido de una transformación no lineal. Es decir, para cualquier vector de entrada $\mathbf{x}$ utilizaron un algoritmo iterativo no lineal para encontrar las representaciones $\mathbf{s}$, aunque la señal estaba modelada mediante $\mathbf{x} = \mathbf{A}\mathbf{s}$. Diseñaron una función general de

[13] Esta selección de matriz de blanqueo se denomina *blanqueo de fase cero* [16], pero es posible utilizar cualquier otra matriz que sea una transformación ortogonal de ${\mathbf{C}_\mathbf{X}}^{-1/2}$.

coste que incorporaba explícitamente el objetivo de obtener pocas componentes activas simultáneamente.

Si en la codificación *sparse* sólo algunas componentes son no nulas, y la mayoría son cero o casi cero, esto significa que la distribución de cada componente es de tipo supergaussiano o leptokúrtica. Por lo tanto, una forma de obtener codificaciones *sparse* es posible también mediante la búsqueda de proyecciones no-gaussianas como en el modelo ICA de la sección anterior.

El hecho de que las componentes individuales de los vectores *sparse* tengan distribuciones supergaussianas puede comprobarse mediante la utilización de alguna medida de *sparseness*. Por ejemplo, una medida muy sencilla de calcular y muy utilizada es la norma L1 del vector:

$$L_1 = \sum_{i=1}^{M} |s_i| \tag{3.56}$$

Cuanto menor sea el valor de la norma L1, menor será el número de componentes activas. Si tomamos el valor esperado de la ecuación:

$$E\{L_1\} = \sum_{i=1}^{M} E\{|s_i|\} \tag{3.57}$$

entonces es posible utilizar $E\{|s_i|\}$ como un estimador de la medida de no-gaussianidad para un vector $\mathbf{s}$ dado, de forma que cuanto menor sea este valor, más supergaussiana será la distribución [79, 77].

Básicamente, para estimar una transformación basada en codificación *sparse* $\mathbf{W}$ se debe especificar alguna medida de *sparseness* y restringir la matriz de transformación $\mathbf{W}$ al tipo lineal. Si el modelo ICA (ecuación 3.52) se mantiene cuando las fuentes (las componentes independientes) tienen distribución *sparse* (supergaussiana), y se elige restringir $\mathbf{W}$ al conjunto de transformaciones no correladas, entonces la estimación de la transformación basada en codificación *sparse* es equivalente al modelo ICA expuesto previamente. Puesto que nuestro interés está centrado en las imágenes, y generalmente las imágenes tienen componentes independientes con distribuciones *sparse* [15, 79, 55], ICA y la codificación *sparse* están estrechamente relacionadas, y los algoritmos a utilizar para las dos transformaciones son equivalentes .

3.4.4. Búsqueda de proyecciones no-gaussianas

En los apartados anteriores se ha visto que tanto la estimación del modelo ICA, como la codificación *sparse*, pueden llevarse a cabo mediante la búsqueda de proyecciones no-gaussianas de los datos. Esto significa que básicamente se está realizado una búsqueda

de proyecciones [14] [77], que se define como la búsqueda de proyecciones de interés de un conjunto de datos mutidimensionales. Las proyecciones de interés son aquellas que se definen como no-normales o no-gaussianas, cuanto menos gaussiana es una distribución, más interesante resulta ser para el modelo ICA.

Los algoritmos de *projection pursuit* requieren el uso de alguna medida de no-normalidad de un vector aleatorio multidimensional. Dado un vector aleatorio multidimensional de varianza unidad, se desea medir con relativa facilidad lo lejos que está su distribución respecto de una distribución gaussiana. Como la información disponible del vector aleatorio viene dada en la forma de muestras (con imágenes, serían diversas imágenes representativas de un mismo objeto), se suelen utilizar estimadores de estas medidas. Existen diversas funciones que realizan esta estimación y se discuten con profundidad en [77]. Aqui, utilizamos G para denotar una de esas funciones; G es una función que mapea el espacio de todas las distribuciones de probabilidad de media cero y varianza unidad en una recta de $\Re$. Se denotará con c el punto en el cual la distribución gaussiana es mapeada:

$$c = G\Big(\frac{1}{\sqrt{2\pi}} exp^{-\frac{x^2}{2}}\Big) \tag{3.58}$$

El punto c divide a la recta real en dos, y es equivalente a la división del espacio de las distribuciones. El conjunto de todas las distribuciones se puede dividir con gran generalidad en dos tipos fundamentales denominados subgaussianos y supergaussianos. Las densidades de probabilidad de las distribuciones subgaussianas son más achatadas que la gaussiana, y las supergaussianas son más picudas, para la mayoría de las funciones G definidas [15]. Las figuras 3.10(a) y 3.10(b) muestran dos ejemplos del aspecto que presentan las distribuciones no-gaussianas (en trazo contínuo) respecto de la distribución normal (en trazo discontínuo).

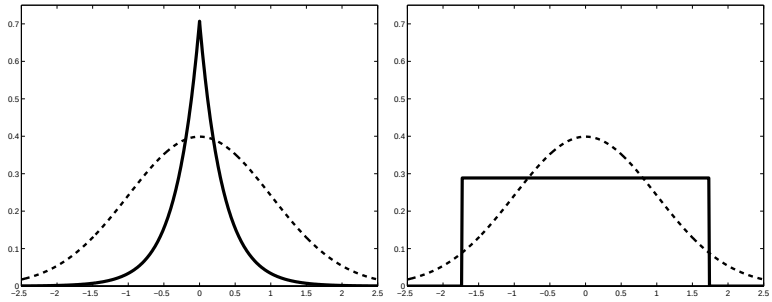
Para encontrar proyecciones no-gaussianas de los datos blanqueados $\mathbf{y}$, se deben buscar extremos de la función $G(\mathbf{b}^T\mathbf{y})$ (con la restricción $\|\mathbf{b}\| = 1$,y por lo tanto $E\{(\mathbf{b}^T\mathbf{y})^2\} = 1$) tales que los máximos y mínimos estén lo más lejos de c como sea posible. Para encontrar el valor de $\mathbf{b}$ bajo esa restricción, cabe utilizar algún algoritmo de optimización, muchos de los algoritmos de ICA utilizan simplemente gradientes ascendentes o descendentes. Por ejemplo, se podría utilizar la siguiente iteración:

$$\mathbf{b} \leftarrow \mathbf{b} \pm \mu \frac{d}{d\mathbf{b}}[G(\mathbf{b}^T\mathbf{y})] \tag{3.59}$$

seguida de una normalización de $\mathbf{b}$ para que tenga norma unidad después de cada paso (esta proyección puede considerarse como una proyección en el conjunto permitido de las $\mathbf{b}$) [81, 126]. Por ejemplo, una medida muy utilizada es la *kurtosis*, que para una variable aleatoria s de varianza unidad es igual a

[14] *projection pursuit*

[15] La división entre subgaussianas y supergaussianas depende de la función elegida G



(a) Distribución supergaussiana, $\kappa > 0$ (*laplaciana*)

(b) Distribución subgaussiana, $\kappa < 0$ (*uniforme*)

Figura 3.10: Ejemplos de distribuciones con distinto grado de no-gaussianidad. Si se utiliza la kurtosis como medida de no-gaussianidad, esta es positiva para las distribuciones supergaussianas, negativa para las subgaussianas y cero para las normales

$$G = \kappa(s) = E\{s^4\} - 3 \tag{3.60}$$

La derivada de la función anterior se puede calcular fácilmente como

$$\frac{d}{d\mathbf{b}}[\kappa(\mathbf{b}^T\mathbf{y})] = E_{\mathbf{y}}\{(\mathbf{b}^T\mathbf{y})^3\mathbf{y}\} \tag{3.61}$$

de forma que se obtiene la siguiente iteración:

1) $\mathbf{b} \leftarrow \mathbf{b} \pm \mu E_{\mathbf{y}}\{(\mathbf{b}^T\mathbf{y})^3\mathbf{y}\}$

2) $\mathbf{b} \leftarrow \mathbf{b}/\|\mathbf{b}\|$

donde el valor esperado se estima a partir de la media de todas las muestras que se disponen.

Una vez se ha localizado una dirección de $\mathbf{b}$ que cumple la condición de máximo no-gaussiano local, es posible construir el correspondiente filtro ICA, $\mathbf{w}$, y la base, $\mathbf{a}$, de la siguiente manera: el filtro ICA es ahora $\mathbf{w}^T = \mathbf{b}^T\mathbf{V}$ ya que se ha realizado primero el blanqueo y posteriormente la búsqueda de la proyección no-gaussiana, siendo la base $\mathbf{a} = \mathbf{V}^{-1}\mathbf{b}$.

Aqui sólo se ha considerado la búsqueda de *una* proyección no-gaussiana. Sin embargo, es sencillo extender el algoritmo de búsqueda a múltiples proyecciones. Esto puede realizarse iterando el algoritmo de sólo una unidad para todos los vectores simultáneamente e ir ortogonalizando en cada paso. Los filtros ICA obtenidos individualmente y las bases pueden luego combinarse en $\mathbf{W}$ y en $\mathbf{A}$ para obtener el modelo ICA matricial.

3.4.5. El algoritmo FastICA

Uno de los inconvenientes del método del gradiente es que se ha de conocer a priori el tipo de fuentes (s) que se están buscando para decidir si realizar el gradiente en modo descendente o ascendente según la medida de no-gaussianidad utilizada. Otra desventaja del gradiente es que, en general, es un método lento y tiene unas propiedades de convergencia bastante pobres [79]. Para solventar dichos problemas, Hyvärinen y Oja desarrollaron un algoritmo denominado *FastICA* [80, 78], debido a su rapidez de ejecución y convergencia, que se desarrolla a continuación.

Recordemos que estamos buscando un extremo de $E\{G(\mathbf{b}^T)\mathbf{y}\}$ bajo la restricción que $\|b\|^2 = 1$. Las condiciones Kuhn-Tucker [16] establecen que ese extremo debe satisfacer

$$\frac{d}{d\mathbf{b}}\Big(E\{G(\mathbf{b}^T)\mathbf{y}\} - \lambda(\|b\|^2 - 1)\Big) = 0 \tag{3.62}$$

realizando la derivada anterior y haciendo el cambio de variable $\beta = 2\lambda$ se obtiene

$$E\{\mathbf{y}G(\mathbf{b}^T\mathbf{y})\} - \beta\mathbf{b} = 0 \tag{3.63}$$

donde β es una constante que puede evaluarse,

$$\beta = E\{\mathbf{b}_0^T\mathbf{y}G(\mathbf{b}_0^T\mathbf{y})\} \tag{3.64}$$

siendo $\mathbf{b}_0$ el valor de $\mathbf{b}$ en el extremo.

Resolviendo la ecuacion 3.63 por el método de Newton, se obtiene la matriz Jacobiana $J(\mathbf{b})$ como

$$J(\mathbf{b}) = E\{\mathbf{y}\mathbf{y}^T G'(\mathbf{b}^T\mathbf{y})\} - \beta\mathbf{I} \tag{3.65}$$

El método de Newton necesita la inversión de la matriz anterior, pero esto se realiza de forma más sencilla convirtiendo la matriz en una de tipo diagonal mediante la aproximación[17]

$$E\{\mathbf{y}\mathbf{y}^T G'(\mathbf{b}^T\mathbf{y})\} \approx E\{\mathbf{y}\mathbf{y}^T\}E\{G'(\mathbf{b}^T\mathbf{y})\} = E\{G'(\mathbf{b}^T\mathbf{y})\}\mathbf{I} \tag{3.66}$$

Por tanto, la solución por el método de Newton viene dada por

$$\mathbf{b} \leftarrow \mathbf{b} - J(\mathbf{b})^{-1}J(\mathbf{b}) = \mathbf{b} - \frac{E\{\mathbf{y}G(\mathbf{b}^T\mathbf{y})\} - \beta\mathbf{b}}{E\{G'(\mathbf{b}^T\mathbf{y})\} - \beta} \tag{3.67}$$

$$\mathbf{b} \leftarrow \mathbf{b}/\|\mathbf{b}\| \tag{3.68}$$

[16] Se trata de condiciones necesarias de optimización que constituyen la generalización de las dadas por Lagrange para problemas con restricciones de desigualdad [8]

[17] Esta aproximación es válida debido a que los datos se han blanqueado previamente

Naturalmente, el valor de β es desconocido, pero si multiplicamos ambos lados de la ecuación 3.69 por $\beta - E\{G'(\mathbf{b}^T\mathbf{y}\}$ la iteración anterior se reduce a

$$\mathbf{b} \leftarrow E\{\mathbf{y}G(\mathbf{b}^T\mathbf{y}\} - \mathbf{b}E\{G'(\mathbf{b}^T\mathbf{y})\} \tag{3.69}$$

$$\mathbf{b} \leftarrow \mathbf{b}/\|\mathbf{b}\| \tag{3.70}$$

3.4.6. Funciones utilizadas en la estimación del modelo ICA

Existen diversos algoritmos para estimar las componentes independientes y los coeficientes de la matriz mezcla según la función de coste, G, elegida.

El enfoque más simple es el visto anteriormente, es decir, el utilizar una medida de no gaussianidad. Según el teorema del límite central, la suma de variables aleatorias independientes no gaussianas está más cerca de la gaussianidad que las variables aleatorias originales. Por tanto, una combinación lineal de las variables observadas presentará mayor no-gaussianidad que las variables fuente, esto es, las componentes independientes. La no gaussianidad se suele cuantificar mediante momentos de orden superior tales como la kurtosis (definida en 3.60), aunque es una medida no muy robusta si existen puntos *outliers*[18] en los datos originales. Por ejemplo, si se disponen 1000 valores de una variable aleatoria (de media cero y varianza unidad) y uno de sus valores es igual a 10, en este caso, la kurtosis es igual al menos a $10^4/1000 - 3 = 7$, lo que significa que un solo valor hace que el valor de la kurtosis sea elevado. El valor de la kurtosis puede depender sólo de una pocas observaciones, las cuales, en algunas ocasiones, pueden ser observaciones erróneas o irrelevantes. En otras palabras, la kurtosis es sensible a los puntos *outliers* y por lo tanto, no es una medida de no gaussianidad muy robusta.

Por otro lado, existen otras medidas de no gaussianidad que pueden funcionar mejor que la kurtosis en algunas ocasiones, como es el caso de la negentropía. Las propiedades de la negentropía son contrapuestas a las de la kurtosis: es robusta pero, computacionalemente compleja. La negentropía es una medida de entropía: es siempre no negativa y es cero para las variables gaussianas (puesto que una variable gaussiana tiene la entropía mayor entre todas las variables aleatorias de igual varianza). Básicamente, la negentropía se define como una versión normalizada de la entropía:

$$N(\mathbf{x}) = H(\mathbf{x}_{\mathbf{gauss}}) - H(\mathbf{x}) \tag{3.71}$$

donde $\mathbf{x}_{\mathbf{gauss}}$ es una variable aleatoria gaussiana y de igual covarianza que $\mathbf{x}$, y H es la entropía de un vector aleatorio $\mathbf{x}$ con densidad de probabilidad $p(\mathbf{z})$,

$$H(\mathbf{x}) = -\int p(\mathbf{z}) \log p(\mathbf{z}) d\mathbf{z} \tag{3.72}$$

[18] Puntos muy alejados del resto de puntos, normalmente correspondientes a observaciones erróneas.

Un método clásico de aproximar la negentropía es utilizar momentos de orden superior, que para el caso de variables unidimensionales toma la forma de la siguiente expresión:

$$N(x) \approx \frac{1}{12} E\{x^3\}^2 + \frac{1}{48} \kappa(y)^2 \tag{3.73}$$

Otra forma de estimar el modelo ICA es utilizar el principio de máxima verosimilitud (estimación ML[19]). En el método ML, se seleccionan aquellos parámetros que proporcionan los valores de probabilidad de las observaciones más altos. Si no existe información a priori a tener en cuenta, el principio ML se convierte en el método MAP[20]. El algoritmo ICA más popular basado en el método ML es el algoritmo de Bell-Sejnowski, también conocido como Infomax [15].

Por último, la estimación ICA también puede realizarse mediante el uso de métodos tensoriales. Entre estos algoritmos, los más extendidos son los algoritmos FOBI [32] y JADE[34].

El algoritmo FastICA puede utilizarse bajo todos los enfoques previamente considerados (diferentes funciones de coste como no-gaussianidad, máxima verosimilitud, información mutua y métodos tensoriales), utilizando el algoritmo iterativo descrito en la ecuación 3.69. Inicialmente, el algoritmo FastICA fue propuesto con la kurtosis como elección de la función de coste [80]. Otras funciones de coste se definen en [78] y [79], siendo las opciones más robustas funciones de coste no polinomiales tales como $\log\cosh$ o $\exp(-y^2)$. Contrariamente a la mayoría de los algoritmos, en FastICA la elección de la función de coste no restringe con gran severidad el tipo de componentes independientes que se obtienen. La elección de G sólo es importante si se desea optimizar el funcionamiento del algoritmo en algún modo especial.

[19] ML, *Maximum likelyhood*
[20] *Maximum a posteriori*

3.4.7. Areas de aplicación de ICA

Una de las posibles aplicaciones de ICA es su uso como método de extracción de características en imágenes, y es el marco considerado dentro de la presente tesis. En el capítulo 5 se describen las ventajas e inconvenientes que presenta el modelo ICA en comparación con el método clásico de PCA en aplicaciones de reconocimiento visual de objetos. Estudios previos muestran resultados contradictorios acerca del funcionamiento de ambos métodos bajo el mismo contexto. Una de las principales aportaciones de esta tesis es determinar el porqué de estos resultados contradictorios y aclarar bajo qué circunstancias ICA puede superar las tasas de éxito en el reconocimiento de PCA.

Sin embargo, las aplicaciones de ICA más conocidas están dentro del campo del procesado de señales: biomédicas, sonoras y de telecomunicaciones, por mencionar algunas. La actividad cerebral se suele medir mediante electroencefalogramas (EEG), magnetoencefalogramas (MEG) o imágenes de resonancia magnética (fMRI). Estas señales pueden considerarse como mezclas de diferentes actividades físicas en el cerebro y otras fuentes de ruido externas a él. ICA puede resolver con éxito el extraer las diferentes fuentes de señales multidimensionales. De forma similar, la separación de señales de voz de diferentes canales, grabadas en diferentes localizaciones, con ruido y retrasadas temporalmente, es otro problema de BSS[21] que ICA resuelve con facilidad. En telecomunicaciones, una línea de transmisión tiene que dividirse entre varios usuarios. La técnica CDMA[22] es una forma moderna de conseguirlo: cada usuario tiene un código individual que distingue su señal de las de los demás mientras las señales son mezcladas durante la transmisión.

Otra de las aplicaciones más relevantes de ICA, es su aplicación en el campo de la Neurociencia Computacional o Neuroinformática. Desde 1996, diversos científicos ([16][127][148], entre otros) sugieren que un tipo de células neuronales de la corteza visual primaria (zona V1) denominadas *simple-cells* pueden ser descritas computacionalmente mediante el Análisis de Componentes Independientes [79] de sus entradas naturales.

Las imágenes naturales pueden definirse como imágenes fotográficas captadas de nuestro entorno natural. Por ejemplo, fotografías de paisajes naturales como bosques, ríos, praderas y animales salvajes dentro de su habitat natural forman parte de la clase imágenes naturales. Las figuras 3.11(a) y 3.11(b) ilustran el concepto de imágenes naturales. La estructura estadística de las imágenes naturales es importante por dos razones fundamentales:

- Es necesaria para cualquier tipo de procesado inteligente de las imágenes. Por ejemplo, los métodos para la compresión de las imágenes, la eliminación de ruido e la

[21] *Blind Source Separation*
[22] *Code Division Multiple Access*

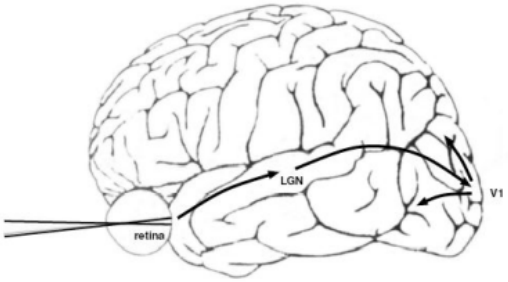
interpolación (zooming) dependen explícita o implícitamente de la estructura estadística inherente en las imágenes.

- Cualquier tipo de sistema visual (bien el de una máquina o bien el de un ser vivo) debe basarse en la estructura de los datos con los que está trabajando. Y por tanto, entender las estadísticas de las imágenes naturales es crucial para entender la corteza visual del cerebro.

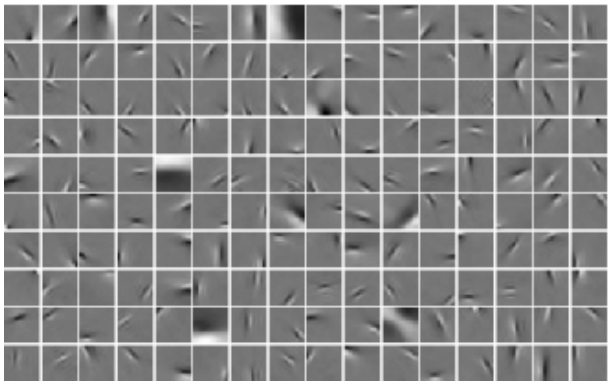
Actualmente, la técnica ICA proporciona un modelo estadístico fundamental para las imágenes naturales, mediante el modelado de las imágenes como superposiciones lineales de imágenes *base* mediante coeficientes o pesos no-gaussianos e independientes [127]. Cuando las imágenes *base* son optimizadas para que el modelo concuerde con los datos de entrada, estas imágenes *base* se convierten en características paso-banda, orientadas y localizadas (ver figuras 3.11(d) y 3.11(e)). Dichas características han sido popularizadas en el campo del procesamiento de imágenes bajo el nombre de *wavelets*. Y además, dicha descomposición es muy semejante a la representación de los campos receptivos de las neuronas *simple-cells* de la corteza visual primaria de los mamíferos.

(a) Imagen natural

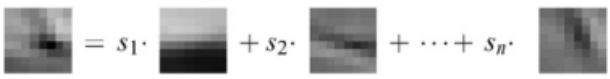
(b) Imagen natural



(c) Localización de la zona V1 (sistema visual primario) en el cerebro.



(d) Campos receptivos de las neuronas de la zona V1.



(e) Modelo ICA aplicado a las imágenes naturales

Figura 3.11: ICA en Neuroinformática.

3.5. Proyecciones aleatorias, RP

El método de proyecciones aleatorias, conocido en la literatura como *Random Projection* (RP), consiste en proyectar los datos originales ($\mathbf{X} \in \Re^{M \times N}$) de dimensión muy alta ($M >> N$) en un espacio de menor dimensión utilizando para ello una matriz de números aleatorios $\mathbf{R} \in \Re^{K \times M}$ tal que $K << M$, y cuyos vectores columna que la forman tienen módulo unidad. Con esta transformación los datos proyectados forman una matriz de dimensiones $K \times N$,

$$\mathbf{P} = \mathbf{R}\mathbf{X} \tag{3.74}$$

La eficacia del método RP se establece en el lema de Johnson-Lindenstrauss [86]: *si un conjunto de N puntos pertenecientes a un espacio vectorial de elevada dimensión (M) se proyectan sobre un espacio aleatoriamente seleccionado de dimensión considerablemente alta (K), donde $K >= O(logN/\epsilon^2)$, entonces las distancias entre puntos son preservadas de forma aproximada (la distorsión tiene un cota máxima de $1 \pm \epsilon$ para cualquier $0 < \epsilon < 1$)* . Una demostración simple de este lema puede consultarse en [39].

La transformación RP es computacionalmente muy simple: construir la matriz aleatoria $\mathbf{R}$ y proyectar la matriz de datos sobre ella. Sin embargo, la ecuacion 3.74 no es estrictamente una proyección porque la matriz $\mathbf{R}$ no es generalmente ortogonal. La transformación lineal descrita en 3.74 puede causar distorsiones significantes en los datos si $\mathbf{R}$ es no ortogonal, aunque aproximadamente se puede considerar que sí es ortogonal basandose en un resultado presentado por Hecht-Nielsen [73]: *en un espacio de elevada dimensión, existen más direcciones casi-ortogonales que no ortogonales.* Por lo tanto, vectores de direcciones aleatorias están con una probabilidad alta lo suficientemente cerca de la ortogonalidad y equivalentemente $\mathbf{R}^T\mathbf{R}$ se aproxima a una matriz identidad [23].

Otra posibilidad consiste en forzar la ortogonalidad de la matriz aleatoria[38]. De esta forma, los pasos a seguir para la obtención de la matriz de transformación $\mathbf{R}$ son:

- Calcular aleatoriamente cada elemento de la matriz $\mathbf{R}$ como un valor de la distribución $N(0, 1)$.
- Ortogonalizar las K filas de la matriz utilizando el método Gram-Schmidt.
- Normalizar las filas de la matriz a 1.

Originalmente, RP se ha utilizado como etapa previa a PCA para reducir la dimensión original de los datos cuando la matriz de covarianzas a utilizar para la descomposición SVD es de dimensión muy elevada; pero también se puede utilizar directamente para

[23] $\mathbf{R}^T\mathbf{R} \in \Re^{M \times M}$.

disminuir la dimensión de los datos antes de utilizar el clasificador. Existen resultados experimentales del funcionamiento de RP en problemas de reducción de la dimensionalidad en diversas aplicaciones de reconocimiento de patrones:

- Papadimitriou *et al.* [131] utilizaron RP en el preprocesado de datos de texto escrito para la tarea de catalogación y clasificación de textos escritos mediante LSI (*Latent Semantic Indexing*). En su trabajo, RP es utilizado para acelerar el algoritmo LSI mediante la reducción de la dimensión de los datos en la etapa previa a la utilización de LSI. De forma similar, Kurimo [95] aplica RP para el indexado de documentos de audio, previamente a la utilización de LSI y SOMs (*Self-Organizing Maps*).

- Dasgupta [38] utiliza RP junto con EM (*Expectation-Maximization*) para modelar distribuciones gaussiana de muy alta dimensión. Y aplica su algoritmo para el indexado y catalogación de textos escritos a mano, obteniendo muy buenos resultados.

- Kaski [88] presenta resultados experimentales obtenidos al utilizar RP en el contexto de un sistema de organización de documentos de texto mediante SOMs. Sus resultados muestran que RP necesita de un número de dimensiones moderado para producir una buena transformación. En este caso, los resultados son tan buenos como los producidos por PCA, y casi tan buenos como aquellos obtenidos con los vectores de datos originales.

- Bingham y Mannila [21] presentan los resultados obtenidos al realizar una comparativa entre RP con distintas selecciones de aleatoriedad y otros métodos de reducción de la dimensión (PCA y la transformada del coseno discreta), utilizando datos obtenidos de catalogación de textos e imágenes.

- Más recientemente, Fradkin y Madigan [62], evalúan RP en el contexto del aprendizaje supervisado. En particular, RP se compara con PCA utilizando diversos tipos de datos y algoritmos de clasificación, concluyendo que aunque RP proporciona unos resultados de clasificación ligeramente peores que los de PCA, sus ventajas computacionales le hacen atractivo para ciertas aplicaciones.

En la presente tesis, se plantea como novedad la utilización de RP como método de extracción de características para la tarea de reconocimiento de objetos basado en la apariencia. En el siguiente capítulo, se presentarán los resultados de clasificación obtenidos con RP, algo inferiores a los de PCA pero bastante prometedores dada la sencillez computacional del método.

3.6. Factorización de Matrices No Negativas, NMF

La factorización de matrices no negativas[24] es una técnica de extracción de características de muy reciente desarrollo [96] [129] que intenta obtener una representación local (o *parts-based*) de datos no negativos. Dada una matriz de datos no negativa $\mathbf{X}$, NMF intenta obtener su factorización en dos matrices no negativas $\mathbf{W}$ y $\mathbf{H}$:

$$\mathbf{X} \approx \mathbf{WH} \tag{3.75}$$

La restricción de no negatividad produce que la representación sea puramente aditiva (no permitiendo sustracciones), en contraste con el resto de representaciones lineales vistas anteriormente como PCA o ICA.

3.6.1. El modelo NMF

NMF es un tipo de representación lineal y no negativa de un conjunto de datos. Siguiendo la notacion utilizada en el apartado 3.2, supongamos que nuestros datos consisten en N medidas o vectores de M variables escalares no negativas, esto es, los datos se pueden agrupar en la matriz $\mathbf{X} \in \Re^{M \times N}$, siendo cada columna de la matriz una medida o vector $\mathbf{x}_j$, donde $j = 1, \ldots, N$. Una aproximación lineal de los datos viene dada por:

$$\mathbf{x}_j \approx \sum_{i=1}^{K} \mathbf{w}_i h_{ij} = \mathbf{W}\mathbf{h}_j \tag{3.76}$$

donde $\mathbf{W}$ es una matriz de tamaño $M \times K$ que contiene a los vectores base $\mathbf{w}_i$ en sus columnas. Los K vectores base $\mathbf{w}_i$ pueden considerarse como los bloques descriptores de los datos, y el vector de coeficientes $\mathbf{h}_j$ (de dimensión K) informa de la mayor o menor importancia de cada bloque descriptor en el vector de datos original $\mathbf{x}_j$.

La ecuación 3.76 se puede expresar también matricialmente como:

$$\mathbf{X} \approx \mathbf{WH} \tag{3.77}$$

donde cada columna de $\mathbf{H}$ contiene al vector de coeficientes $\mathbf{h}_j$ correspondiente al vector de datos $\mathbf{x}_j$. A la vista de la expresión 3.77, se puede observar que una representación lineal de los datos es simplemente una factorización de la matriz de datos. Las técnicas de PCA, ICA y NMF pueden considerarse todas ellas como factorizaciones de matrices con diferentes elecciones en la función de coste o en las restricciones que imponen en la representación.

Mientras que PCA e ICA no restringen los signos de los valores de $\mathbf{W}$ y $\mathbf{H}$, NMF impone que todos los valores de ambas matrices sean no negativos, lo que significa que

[24] En la literatura se utiliza el término NMF para el método *Non-negative Matrix Factorization*

los datos son descritos utilizando sólo componentes aditivas. Esta restricción impuesta por NMF está motivada principalmente por las siguientes dos consideraciones:

- En la mayoría de los sistemas físicos reales las cantidades implicadas no pueden ser negativas. En tales casos, puede ser difícil interpretar los resultados de PCA e ICA [129][134].
- La no negatividad surge en los sistemas locales o basados en partes puesto que las partes, generalmente, se combinan aditivamente (no mediante substracción) para formar un todo [96].

En la representación NMF, dada la matriz de datos $\mathbf{X}$, las matrices $\mathbf{W}$ y $\mathbf{H}$ se definen como aquellas matrices no negativas que minimizan el error de reconstrucción entre $\mathbf{X}$ y $\mathbf{WH}$. Se han propuesto diversas funciones de error [68][96][129], siendo quizás la más utilizada la función de error cuadrático (distancia euclídea):

$$E(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{WH}\|^2 = \sum_{i,j} (x_{ij} - (\mathbf{WH})_{ij})^2 \tag{3.78}$$

Paatero y Tapper [129] proporcionaron un algoritmo basado en gradiente para la optimización de la ecuación 3.78, mientras que Lee y Seung [96] presentaron un algoritmo multiplicativo más sencillo de implementar.

Existen pocos trabajos teóricos sobre las propiedades de la representación NMF, siendo la excepción recientemente el trabajo de Donoho y Stodden [45]. Sin embargo, el éxito de la representación NMF radica en su capacidad de obtener características significativas de colecciones de datos biológicos reales. Concretamente Lee y Seung [96] mostraron que cuando $\mathbf{X}$ es una colección de imágenes de caras[25], la representación NMF genera vectores base que muestran al visualizarlos como imágenes las características intuitivas propias de las caras, esto es, ojos, boca, nariz, etc. A pesar de este éxito de NMF en la obtención de características locales de imágenes de caras, existen conjuntos de imágenes en los que NMF no proporciona la descomposición en partes que se esperaba según los resultados de Lee y Seung [96] (ver Fig.2.9(b)). En concreto, Li *et al.* [100] mostraron que cuando se aplica NMF a la base de datos de imágenes de caras ORL [3], los vectores obtenidos proporcionan una representación global más que local. La figura 3.12 muestra algunas imágenes de la base de datos de caras ORL y en la figura 3.13 se reproducen los resultados de Li *et al.* [100]: se muestran los vectores base de NMF obtenidos con las imágenes de caras de la ORL y se aprecia que las características obtenidas no son las características locales propias de las caras presentadas por Lee y Seung [96]. La diferencia del comportamiento de la representación NMF en estas dos librerías de caras se atribuye a la mejor o peor alineación entre las caras de la base de datos [100].

[25]Concretamente Lee y Seung [96] en sus experimentos con caras utilizaron la librería CBCL [2].

Figura 3.12: Algunas imágenes de la base de datos de caras ORL.

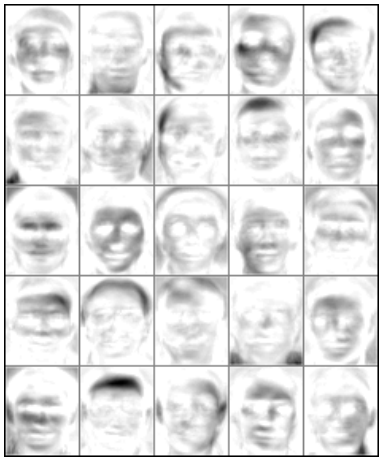
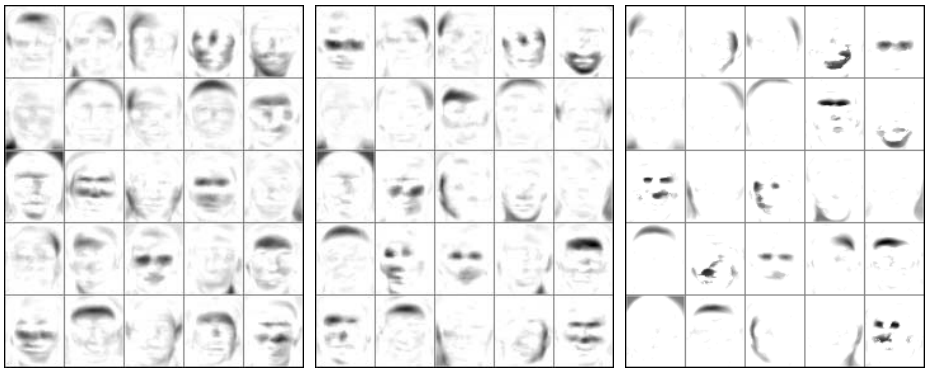


Figura 3.13: Características obtenidas con la base de datos de caras ORL utilizando el algoritmo clásico de NMF. Se obtienen características globales y no locales de las imágenes de entrenamiento.



(a) sW = 0.5 (b) sW = 0.6 (c) sW = 0.7

Figura 3.14: Características obtenidas con la base de datos de caras ORL utilizando el algoritmo NMFSC para distintos valores de sW

Recientemente, Hoyer [75] ha propuesto un nuevo algoritmo de cómputo de la representación NMF que incorpora el concepto de *sparseness* visto en el apartado 3.4.3 y que permite mejorar la obtención de vectores de características locales de los datos originales. Las figuras 3.14(a), 3.14(b) y 3.14(c) muestran el resultado de aplicar este nuevo método sobre la librería de caras ORL: para ciertos valores de *sparseness* la representación obtenida es de tipo local.

3.6.2. NMF con restricciones de codificacion *sparse*

El objetivo es forzar que el algoritmo NMF encuentre soluciones con los grados de *sparseness* deseados. La primera cuestión a resolver es: ¿qué debe ser *sparse*, los vectores base $\mathbf{W}$ o los coeficientes $\mathbf{H}$?

La pregunta anterior no tiene una respuesta general, puesto que la elección depende de la aplicación específica en cuestión. Además, sólo trasponiendo la matriz de datos original se intercambian los valores de $\mathbf{W}$ y $\mathbf{H}$, por lo que la elección de cuál de ellas restringir debe ser hecha por el usuario en cada caso concreto. Por ejemplo, un doctor analizando patrones de enfermedades puede asumir que la mayoría de las enfermedades son raras (y por lo tanto, *sparse*) pero que cada enfermedad puede causar un número elevado de síntomas. Si las filas de la matriz de datos original están formadas por los distintos síntomas, y los distintos sujetos o pacientes se ordenan en las columnas, en estas condiciones los coeficientes deberían ser *sparse* y los vectores base no necesariamente.

Por otro lado, si intentamos extraer características útiles de un conjunto de imágenes, puede tener sentido que ambas, $\mathbf{W}$ y $\mathbf{H}$, sean *sparse*, lo que significaría que cualquier objeto está presente en sólo unas pocas imágenes y afecta sólo a una pequeña parte de la

imagen.

Las consideraciones anteriores conducen a la definición formal de NMFSC[26]:

- Dada una matriz no negativa $\mathbf{X}$ de tamaño $M \times N$, se desea obtener dos matrices no negativas $\mathbf{W}$ y $\mathbf{H}$ de dimensiones $M \times K$ y $K \times N$ tal que:

$$E(\mathbf{W}, \mathbf{H}) = ||\mathbf{X} - \mathbf{WH}||^2 \tag{3.79}$$

  es mínimo, bajo las restricciones opcionales:

$$\begin{aligned} sparseness(\mathbf{w}_i) &= S_w, \forall i \\ sparseness(\mathbf{h}_i) &= S_h, \forall i \end{aligned} \tag{3.80}$$

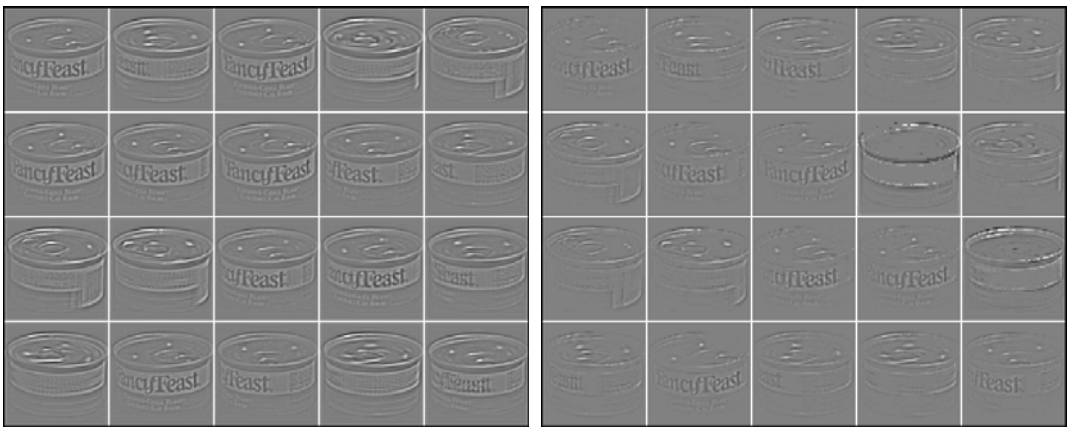
  donde $\mathbf{w}_i$ es la columna i de $\mathbf{W}$ y $\mathbf{h}_i$ es la fila i de $\mathbf{H}$. K denota el número de componentes, y S_w y S_h son los valores de *sparseness* de $\mathbf{W}$ y $\mathbf{H}$, respectivamente. Estos tres parámetros (K, S_w y S_h) son ajustados por el usuario, según las especificaciones del problema.

En la definición anterior no se restringen las escalas de $\mathbf{w}_i$ y $\mathbf{h}_i$. Sin embargo, como se cumple que $\mathbf{w}_i\mathbf{h}_i = (\mathbf{w}_i\lambda)(\mathbf{h}_i/\lambda)$, existe libertad para fijar la norma de ambos vectores. En el algoritmo original de NMFSC [75] se elige arbitrariamente fijar la norma L2 de $\mathbf{h}_i$ a la unidad.

3.6.3. Algunas consideraciones sobre NMFSC

La técnica de NMFSC presenta propiedades particulares que la hacen potencialmente muy adecuada para aplicaciones de reconocimiento de objetos. Ajustando adecuadamente el valor de *sparseness*, se puede conseguir extraer características locales de las imágenes, con independencia del tipo de imágenes utilizado (a diferencia del método NMF estándar). En las figuras 3.15(a) y 3.15(b) se muestran los resultados obtenidos sobre las imágenes de uno de los objetos de la librería COIL-100 con ambos métodos (en concreto, el objeto 29). Es posible apreciar cómo NMFSC extrae características locales (resalta detalles locales de las imágenes) mientas que NMF extrae únicamente características globales. En la figura 3.15(b) se aprecia como en algunas imágenes resaltan letras, en otras la anilla superior o simplemente la forma cilíndrica del objeto. Al contrario, si se utiliza el algoritmo de Lee and Seung [96] sobre el mismo objeto, no se obtienen estas representaciones locales, tal como se muestra en la figura 3.15(a).

[26] En el apéndice B se describe con más detalle el algoritmo NMFSC.



(a) NMF (b) NMFSC, $S_w = 0{,}7 S_h = 0$

Figura 3.15: Aplicación de NMF y NMFSC a un mismo conjunto de imágenes (objeto nº 29 de la COIL-100). Se aprecia cómo solamente NMFSC es capaz de extraer o resaltar características locales de las imágenes.

Desde el punto de vista práctico, la posibilidad de extracción de características locales permite obtener claras ventajas en aplicaciones de reconocimiento de objetos. A continuación se detallan algunas situaciones en las que podría resultar beneficiosa la aplicación de esta técnica:

- La primera situación práctica en la que la aplicación de NMFSC puede resultar adecuada es el reconocimiento de objetos en presencia de oclusiones: cuando existen oclusiones, no es posible reconocer características globales en los objetos; pero sí es posible reconocer características locales o detalles, que pueden llevar un aidentificación correcta. Aunque el reconocimiento de objetos en presencia de oclusiones no es el objetivo de la presente tesis, se han realizado experimentos que se mostrarán en apartados posteriores y que permiten comprobar la eficacia del método NMFSC en tales circunstancias.

- En segundo lugar, las aplicaciones de categorización también son adecuadas para el uso de NMFSC. En tales aplicaciones, el objetivo no es identificar un cierto objeto a partir de su imagen sino identificar la categoría a la que el objeto pertenece. Dos objetos pertenecen a la misma categoría si, aun siendo diferentes, poseen ciertas características comunes (ejemplo: dos tazas distintas; dos automóviles distintos; etc.). En tal situación, las características globales que pueden extraerse con técnicas tradicionales no son del todo eficaces dado que no se busca una similitud completa entre objetos; sin embargo, las características locales que obtiene NMFSC sí que pueden permitir asignar diferentes objetos a una misma categoría en función de

detalles comunes. La categorización de objetos tiene un campo de aplicación muy extenso en sistemas tales como la extracción automática de información a partir de bases de datos de imágenes.

- Por último, NMFSC también puede reportar beneficios en una aplicación común de detección de objetos o localización de objetos en un entorno no estructurado. En estas aplicaciones, si bien las características globales aportan información adecuada para la detección, es posible complementar esa información con características locales, mediante el uso combinado de NMFSC y técnicas clásicas. Disponer de mayor información puede permitir aumentar las tasas de acierto en detección.

La última de las estrategias mecionadas se aplica sobre un entorno real en apartados posteriores de esta tesis. Se emplea NMFSC en una de las fases de un sistema de detección de objetos en entornos no estructurados, en combinación con PCA. Los resultados obtenidos en múltiples experimentos demuestran que la adición de NMFSC permite mejorar las tasas de acierto del sistema en condiciones diversas; particularmente en situaciones en las que la detección se complica por factores tales como presencia de numerosos objetos circundantes, existencia de oclusiones, etc.

Capítulo 4

Arquitectura de reconocimiento de objetos propuesta

Las diversas formas de emplear los métodos de extracción de características para reducir el tamaño de las imágenes se denominan arquitecturas y, clásicamente, se distinguen dos tipos de arquitecturas denominadas universal y objeto.

Este capítulo propone una nueva arquitectura basada en la arquitectura objeto original de Murase [118]. La nueva propuesta permite no sólo detectar orientaciones sino también identificar objetos.

A continuación se detalla su metodología y se presentan los resultados experimentales obtenidos con las arquitecturas clásicas y la propuesta en esta tesis bajo el marco de la técnica de PCA. Estos experimentos se han realizado utilizando dos librerías de objetos de forma libre, una ampliamente conocida y utilizada, COIL-100, y una nueva librería de objetos, UMH-15, generada para la ocasión.

4.1. Subespacios Universal y Objeto

Los sistemas de reconocimiento de objetos basado en la apariencia pueden implementarse utilizando dos tipos de arquitecturas básicas propuestas por Murase [118] y que se conocen como subespacio *universal* (o *global*) y subespacio *objeto*[1].

El subespacio universal está compuesto por los vectores que definen la transformación de extracción de características utilizada, PCA, ICA u otra, obtenidos a partir de un conjunto de entrenamiento de imágenes pertenecientes a *objetos diferentes*.

El subespacio universal es la matriz de transformación $\mathbf{W}$ en la ecuación 3.7 cuando la matriz de datos $\mathbf{X}$ está compuesta por imágenes pertenecientes a objetos diferentes. Para el caso concreto de PCA, el subespacio universal es la matriz $\mathbf{U}$ de la ecuación 3.29; la matriz $\mathbf{U}$ está formada por los vectores propios de la matriz de covarianzas de los datos $\mathbf{C_X}$. En el ejemplo de PCA que se mostró en la figura 3.5 el subespacio universal lo forman las imágenes de la segunda fila (los ocho vectores propios).

El subespacio objeto se obtiene a partir de un conjunto de *imágenes del mismo objeto* en diferentes poses o vistas. De este modo, el subespacio universal se utiliza para discriminar entre objetos, mientras que el subespacio objeto se utiliza para estimar la pose del objeto [118].

La pose y la iluminación de un objeto pueden cambiar según la escena considerada. Si se desea recopilar la máxima información visual posible de un objeto para su posterior reconocimiento, es necesario utilizar varias imágenes de las diferentes vistas (o poses) del objeto, así como diversas imágenes que recojan los distintos grados de luminosidad en los que puede encontrarse el objeto. De este modo, para un objeto determinado t se puede definir el conjunto de imágenes de entrenamiento del objeto como:

$$G_t = \{\mathbf{x_1^t}, \mathbf{x_2^t}, \ldots, \mathbf{x_j^t}, \ldots, \mathbf{x_N^t}\} \tag{4.1}$$

donde cada vector $\mathbf{x_j^t}$ denota el vector de características con la información de cada objeto en el canal de características seleccionado para representar la apariencia del objeto (recordar la definición de espacio imagen del apartado 2.2).

Algunos canales de características típicos son: la imagen en niveles de gris, la imagen en un espacio de color modificado, el histograma de la imagen, la imagen de bordes, transformadas de la imagen (laplaciana, log, wavelets, fourier, ...) o áreas especiales de la imagen agrupadas en ventanas [13][66][124].

En general, para un sistema de reconocimiento compuesto por T objetos diferentes,

[1] En el artículo original de Murase [118] se emplean los términos *universal and object eigenspaces* o también *manifolds* para referirse a los subespacios PCA generados con cada una de las arquitecturas

podemos definir el *conjunto de imágenes universal* como la unión de los conjuntos de imágenes de entrenamiento de todos los objetos:

$$\begin{aligned} G_U \;=\; & \{\mathbf{x_1^1}, \mathbf{x_2^1}, \ldots, \mathbf{x_j^1}, \ldots, \mathbf{x_N^1}\}, \\ & \{\mathbf{x_1^2}, \mathbf{x_2^2}, \ldots, \mathbf{x_j^2}, \ldots, \mathbf{x_N^2}\}, \\ & \vdots \\ & \{\mathbf{x_1^t}, \mathbf{x_2^t}, \ldots, \mathbf{x_j^t}, \ldots, \mathbf{x_N^t}\}, \\ & \vdots \\ & \{\mathbf{x_1^T}, \mathbf{x_2^T}, \ldots, \mathbf{x_j^T}, \ldots, \mathbf{x_N^T}\} \end{aligned} \tag{4.2}$$

donde cada vector $\mathbf{x}_j^t$ denota el vector de características con la información de cada objeto t en el canal de características seleccionado para representar la apariencia del objeto.

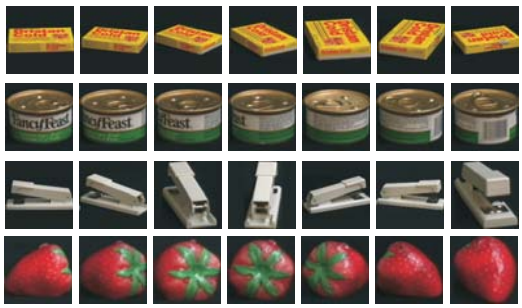
$G_U \;=$ 

Figura 4.1: Ejemplo de un conjunto de imágenes universal con algunos objetos (1, 29, 68 y 75) de la librería COIL-100.

Como se vio en el apartado 2.2, si consideramos que una imagen de un objeto es un punto del espacio imagen, una colección de N imágenes es equivalente a N puntos del espacio imagen. Bajo este enfoque, el reconocimiento de objetos consiste en realizar agrupaciones de puntos en el espacio imagen pertenecientes al mismo objeto, pero debido al tamaño considerable de las imágenes y sobre todo si se trabaja con muchos objetos diferentes, es necesario realizar un proceso de extracción de características para comprimir la información de las imágenes de los objetos. Si esta transformación se obtiene a partir del conjunto de imágenes universal, los vectores fila de la matriz de transformación ($\mathbf{W}$) son los vectores que definen el subespacio universal. La proyección de las imágenes originales en este subespacio será:

$$\mathbf{P} = \mathbf{W}^T\mathbf{X} \tag{4.3}$$

donde la matriz $\mathbf{P}$ recoge en sus columnas la información de las imágenes tras la transformación del subespacio universal.

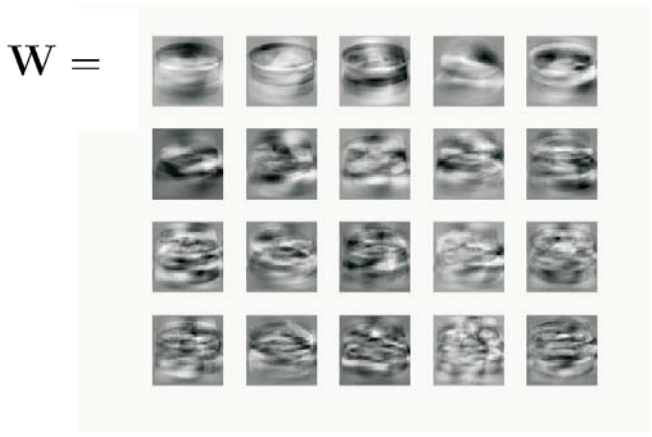


Figura 4.2: Ejemplo de subespacio universal: los 20 primeros vectores PCA obtenidos a partir de las 72 imágenes de cada uno de los cuatro objetos mostrados en la figura 4.1 .

La figura 4.2 muestra un ejemplo de subespacio universal, en concreto, los 20 primeros vectores PCA obtenidos a partir de las 72 imágenes de cada uno de los cuatro objetos mostrados en la figura 4.1. Si se utilizan sólo dos vectores de este subespacio para realizar la transformación, las imágenes originales quedarán reducidas a puntos bidimensionales, tal como se muestra en la figura 4.3. En esta figura se aprecia que simplemente con 2 dimensiones hay tres clases u objetos perfectamente separados (objetos 1, 29 y 75), sólo los patrones del objeto 68 se confunden con el resto de clases. Cabe destacar que se ha representado un caso muy simple, es decir, las imágenes originales de 16384 dimensiones (128×128) han sido reducidas a puntos de 2 dimensiones, y sin embargo, los patrones obtenidos están claramente separados, lo que permite percibir la potencia del método de reconocimiento basado en apariencia utilizando PCA. Cuanto más cercanas están las proyecciones en el subespacio, más correladas están las imágenes en el espacio imagen original. Esta propiedad de los subespacios PCA permite su uso en la tarea de reconocimiento de objetos desde la perspectiva de la visión por computador, donde la correlación es utilizada como medida de similitud entre imágenes.

El subespacio objeto se calcula para cada objeto t, extrayendo los vectores de PCA, ICA, NMF o de cualquier otra transformación, a partir sólo de los datos presentes en el conjunto de imágenes de entrenamiento del objeto. Si estas imágenes son proyectadas en su propio subespacio objeto, generando un vector multidimensional de menor número de componentes, se obtendrá una representación de la trayectoria del objeto en función de su vista (rotación) en el espacio multidimensional de características seleccionado. Esta representación es útil para intentar reconocer la pose del objeto. En la figura 4.4 se muestra una trayectoria en el subespacio objeto del objeto nº 14 de la librería COIL-100, utilizando sólo 3 dimensiones.

El sistema de reconocimiento de objetos propuesto por Murase [118] consistía en utili-

zar el subespacio universal PCA para discriminar entre objetos, es decir, dada una imagen de un objeto desconocido, asociarlo a uno de los objetos del sistema o bien rechazarlo como objeto no perteneciente al sistema (clase rechazo); mientras que el subespacio objeto PCA, se utilizaba posteriormente para estimar la pose o vista del objeto respecto de la cámara.

Nuestra propuesta es utilizar los subespacios objeto también para reconocimiento y no sólo para localización como se decribe detalladamente en la siguiente sección.

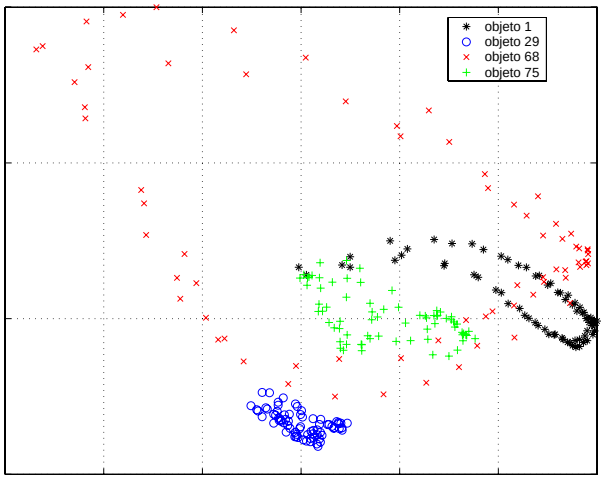


Figura 4.3: Proyecciones ($\mathbf{P}$) del conjunto de imágenes universal de 4 objetos de la COIL-100 (72 imágenes por cada objeto) en un subespacio universal PCA de 2 dimensiones.

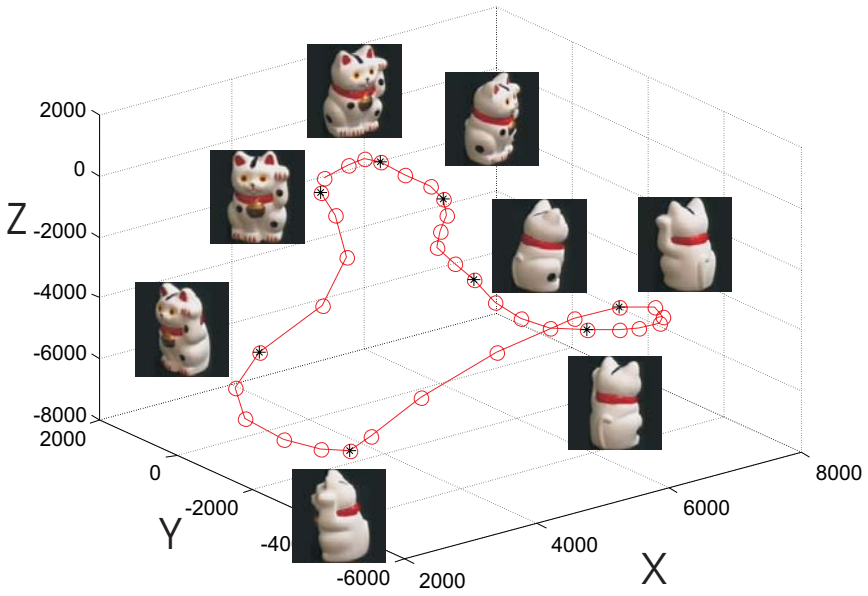


Figura 4.4: Ejemplo de subespacio objeto: trayectoria en el subespacio objeto PCA del objeto nº 14 de la librería COIL-100, utilizando sólo 3 dimensiones

4.2. Arquitectura propuesta: Subespacio Objeto Modificado

Dada una imagen o un conjunto de imágenes pertenecientes a un objeto desconocido, si se desea utilizar los subespacios objeto para identificar a qué clase pertenece el objeto, habría que proyectar dicha imagen o conjunto sobre cada uno de los subespacios objeto de los diferentes objetos presentes en el sistema de reconocimiento. Por ejemplo, si el sistema de reconocimiento estuviese formado por 3 objetos de la COIL-100 (1, 29 y 68), la metodología a seguir para identificar un objeto sería la siguiente:

1 Generar los subespacios objeto a partir del conjunto de imágenes de entrenamiento de cada objeto. Sean G_1, G_{29} y G_{68} estos conjuntos de entrenamiento, y W_1, W_{29} y W_{68} los vectores generadores de los subespacios objeto obtenidos mediante algún método de extracción de características. La figura 4.5 muestra posibles conjuntos de entrenamiento para estos objetos utilizando 36 imágenes por objeto.

2 Proyectar las imágenes de entrenamiento en cada subespacio objeto propio:

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{W}_1^T \mathbf{G}_1 \\ \mathbf{P}_{29} &= \mathbf{W}_{29}^T \mathbf{G}_{29} \\ \mathbf{P}_{68} &= \mathbf{W}_{68}^T \mathbf{G}_{68} \end{aligned} \tag{4.4}$$

Con esto se consigue el modelo de puntos de cada objeto a utilizar por el clasificador. Si la matriz de transformación está formada por sólo 2 vectores, las proyecciones de los objetos en los subespacios objeto serán puntos bidimensionales. La figura 4.6 muestra en gráficos separados cada una de estas proyecciones $\mathbf{P}_1$, $\mathbf{P}_{29}$ y $\mathbf{P}_{68}$ calculadas utilizando las matrices de transformación W_t de PCA.

Los patrones $\mathbf{P}_1$, $\mathbf{P}_{29}$ y $\mathbf{P}_{68}$ son los ejemplos que utilizará el clasificador para identificar a una nueva imagen como perteneciente o no a una de estas tres clases de objetos. Supongamos que el clasificador es el vecino más cercano con distancia euclídea como medida de distancia.

3 Proyectar la imagen o conjunto de imágenes del nuevo objeto (*imágenes de test*), $\mathbf{T}$, en cada uno de los subespacios objeto.

$$\begin{aligned} \mathbf{T}_{W1} &= \mathbf{W}_1^T \mathbf{T} \\ \mathbf{T}_{W29} &= \mathbf{W}_{29}^T \mathbf{T} \\ \mathbf{T}_{W68} &= \mathbf{W}_{68}^T \mathbf{T} \end{aligned} \tag{4.5}$$

Estas proyecciones del nuevo objeto serán la entrada al clasificador que identificará el objeto desconocido como aquél que presenta una distancia mínima entre las nuevas proyecciones y las clases representadas por las proyecciones modelo.

Sin embargo, estos pasos que se acaban de describir no son del todo apropiados para el reconocimiento de objetos. Para ilustrarlo, se muestran los resultados que se obtienen con las imágenes de test correspondientes a los tres objetos de la librería COIL-100 sobre los que se está trabajando.

La figura 4.7 muestra el resultado de proyectar las imágenes de test del primero de los objetos sobre cada uno de los tres subespacios. Cuando la proyección se realiza sobre el subespacio generado a partir de vistas del propio objeto (figura 4.7(a)), los resultados son adecuados: las proyecciones de las imágenes de test se sitúan próximas a las proyecciones de las imágenes de entrenamiento, y por tanto el objeto se identificaría correctamente. Sin embargo, cuando la proyección se realiza sobre un subespacio generado a partir de vistas de un objeto distinto (figuras 4.7(b) y (c)), los resultados son en cierto modo aleatorios: algunas de las proyecciones se sitúan próximas a las proyecciones de las imágenes de entrenamiento utilizadas para generar la vista (lo cual es un mal comportamiento, dado que provocaría identificaciones incorrectas) mientras que algunas otras se sitúan alejadas (lo que indicaría un comportamiento correcto). Esta aleatoriedad se debe a que las direcciones sobre las cuales se proyecta no reflejan la variabilidad propia del objeto cuyas imágenes se están proyectando (porque no se ha utilizado para generar el subespacio).

El mismo comportamiento se observa en las figuras 4.8 y 4.9. El único caso en el que las proyecciones son adecuadas es aquél en el cual el subespacio se ha generado con vistas del mismo objeto cuyas imágenes se proyectan. Particularmente, en la figura 4.9 se aprecia cómo los resultados en los otros casos (figuras 4.9(a) y 4.9(b) son claramente aleatorios.

Para paliar este problema, se propone la utilización de una arquitectura distinta, aunque también basada en la utilización de subespacios objeto. Para distinguir claramente esta arquitectura, se utilizará de ahora en adelante la expresión ***subespacio objeto modificado***.

Para un sistema de reconocimiento de objetos formado por T objetos diferentes, el subespacio objeto modificado consiste en la unión de los subespacios objeto de cada uno de los objetos:

$$\mathbf{W}_{om} = \bigcup [\mathbf{W}_1 \mathbf{W}_2 \dots \mathbf{W}_t \dots \mathbf{W}_T] \tag{4.6}$$

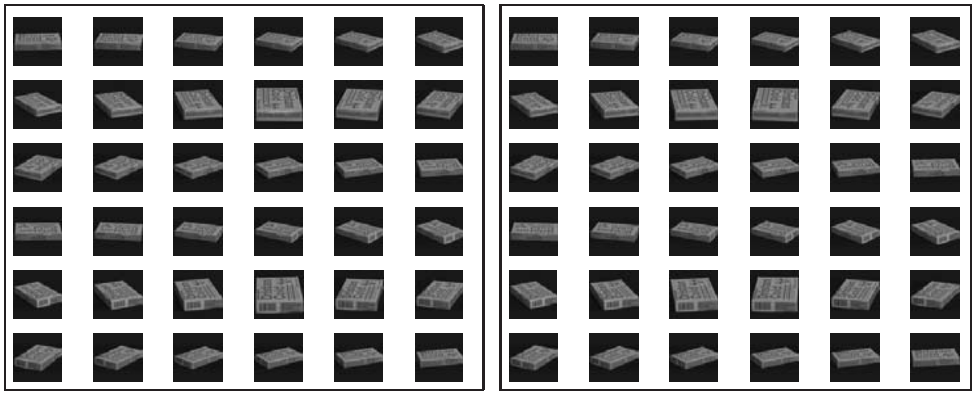
la proyección de las imágenes originales en este subespacio será:

$$\mathbf{P}_{om} = \mathbf{W}_{om}^T \mathbf{X} \tag{4.7}$$

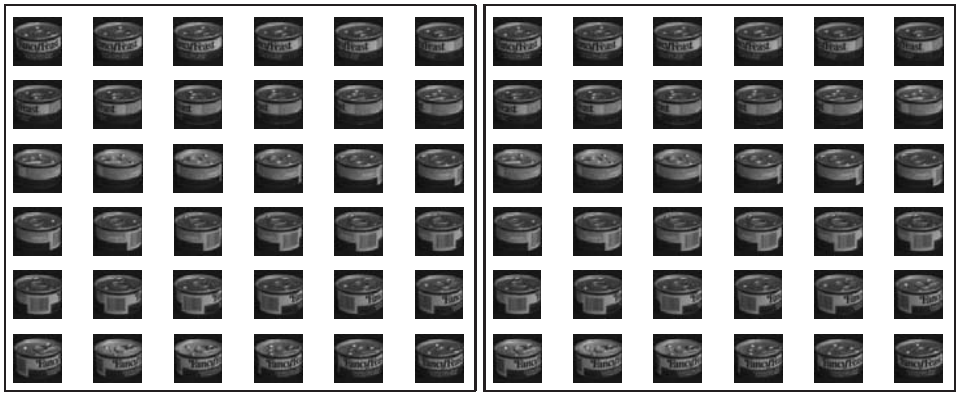
donde la matriz $\mathbf{P}_{om}$ recoge en sus columnas la información de las imágenes tras la transformación.

La figura 4.10 presenta los resultados obtenidos con la utilización del subespacio objeto modificado, sobre los tres mismos ejemplos utilizados anteriormente. Para posibilitar la representación gráfica, se utilizan únicamente tres dimensiones, que se corresponden con tres de las componentes del subespacio. La elección de tales componentes se hace de la forma siguiente: dado que se trabaja únicamente con tres objetos, se elige la componente de máxima varianza de cada objeto, de modo que todos queden representados. En la figura se representan simultáneamente las proyecciones de las imágenes de entrenamiento (utilizadas para generar el subespacio) y de test, para cada uno de los objetos. En todos los casos el comportamiento es correcto, como era esperable: las imágenes de test se intercalan aproximadamente entre las imágenes de entrenamiento.

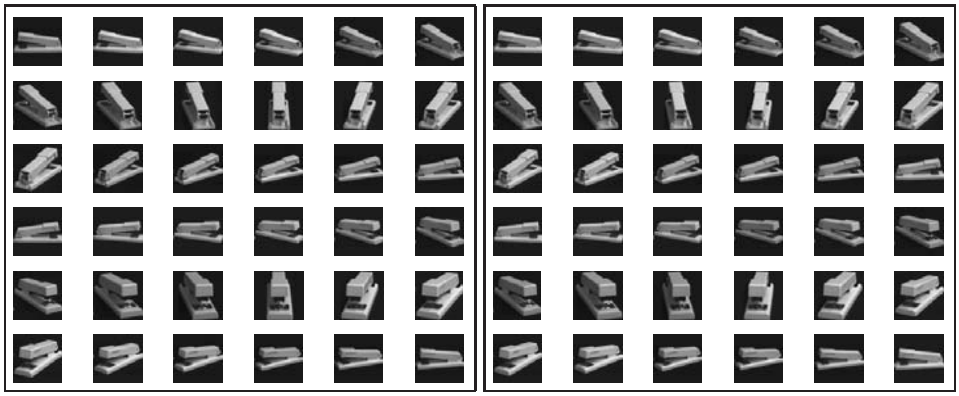
La figura 4.11 muestra un resultado más interesante, el funcionamiento de la arquitectura propuesta para distinguir objetos diferentes (aspecto en el que la utilización directa de los subespacios objeto no ofrece buenos resultados, de acuerdo con lo comentado anteriormente). En este caso, se observa cómo el subespacio objeto modificado sí es válido para distinguir entre objetos. En la figura 4.11(a) se representan con marcadores diferenciados las imágenes de entrenamiento y test de dos objetos diferentes, y es posible comprobar cómo las imágenes de test siempre se clasificarían correctamente (por ejemplo, utilizando como clasificador el método del vecino más cercano) dada su cercanía a imágenes de entrenamiento de la misma clase (del mismo objeto). En la figura 4.11(b), para una mayor claridad, se representan con el mismo marcador las imágenes de entrenamiento y test, y se aprecia claramente la correcta separación entre clases.



(a) Entrenamiento, $\mathbf{G}_1$ (b) Test, $\mathbf{T}_1$

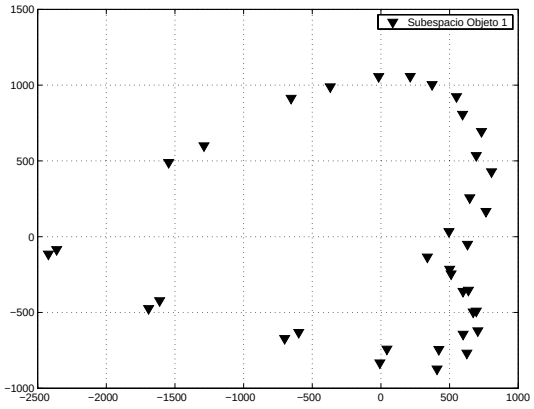


(c) Entrenamiento, $\mathbf{G}_{29}$ (d) Test, $\mathbf{T}_{29}$

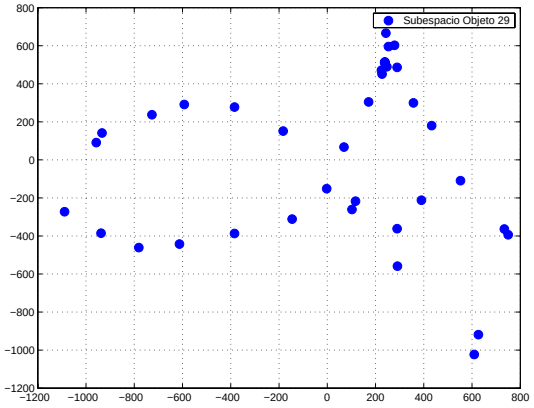


(e) Entrenamiento, $\mathbf{G}_{68}$ (f) Test, $\mathbf{T}_{68}$

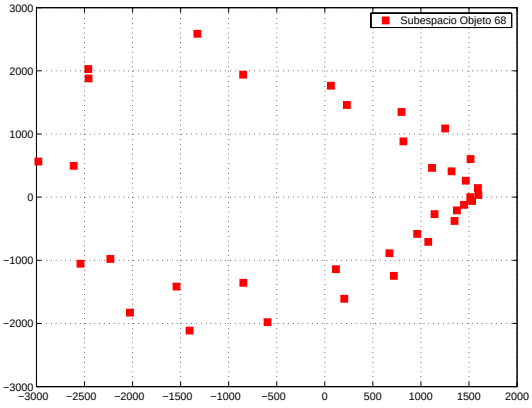
Figura 4.5: Imagenes de los objetos 1, 29 y 68 de la librería COIL-100. Conjuntos de imágenes de entrenamiento: (a) Objeto 1; (c) Objeto 29; (e) Objeto 68. Conjunto de imágenes de test: (b) Objeto 1; (d) Objeto 29; (f) Objeto 68.



(a) $\mathbf{P}_1$

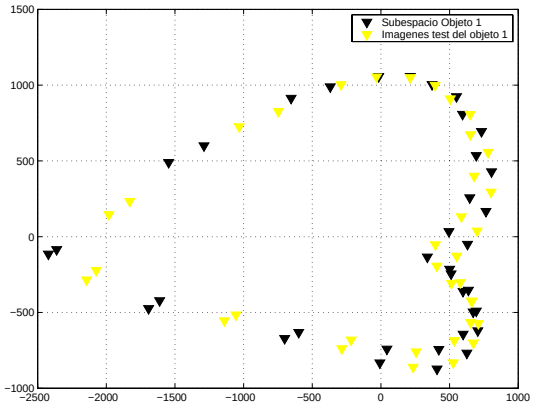


(b) $\mathbf{P}_{29}$

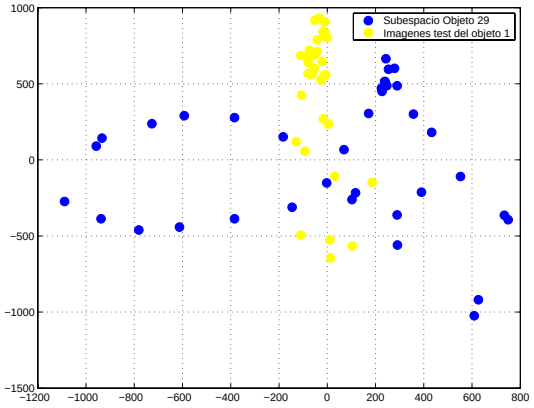


(c) $\mathbf{P}_{68}$

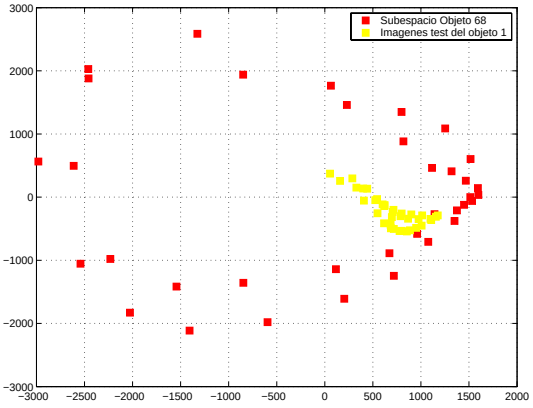
Figura 4.6: Representación bidimensional de los subespacios objeto PCA en el canal I de los objetos 1, 29 y 68 de la librería COIL-100.



(a)

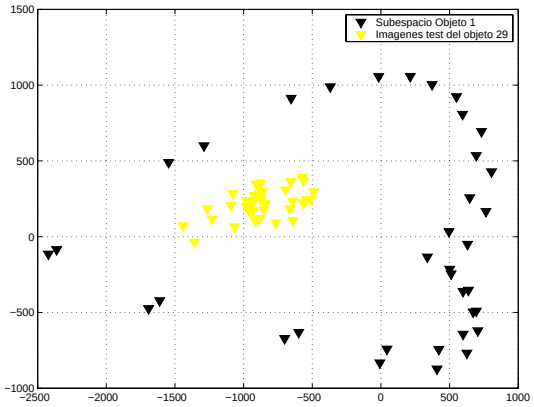


(b)

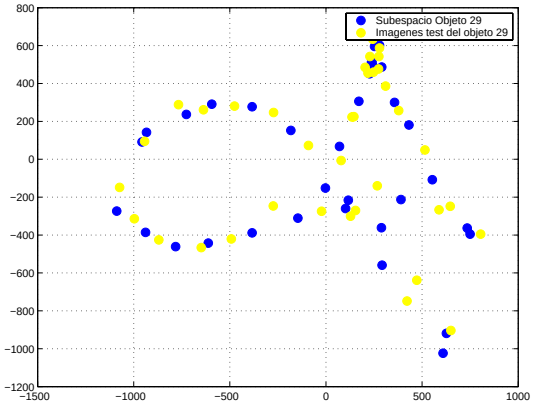


(c)

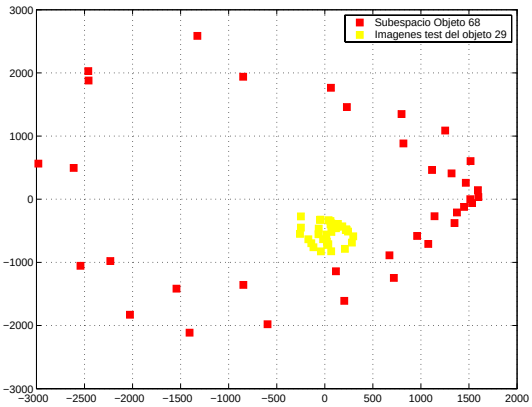
Figura 4.7: Proyección de las imágenes de test del objeto 1 sobre cada uno de los subespacios objeto.



(a)

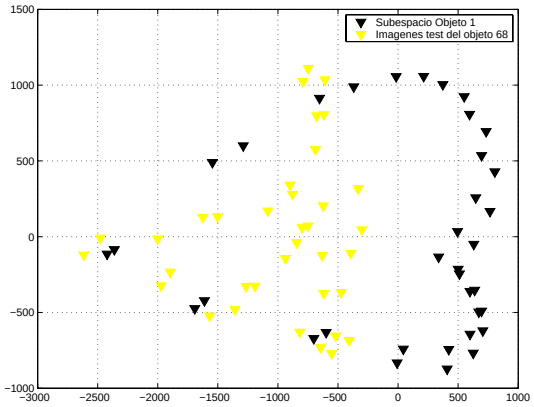


(b)

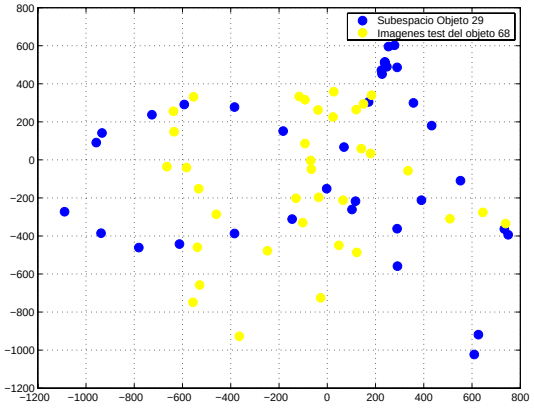


(c)

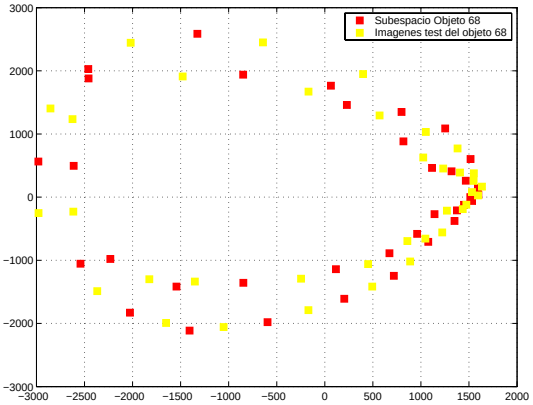
Figura 4.8: Proyección de las imágenes de test del objeto 29 sobre cada uno de los subespacios objeto.



(a)

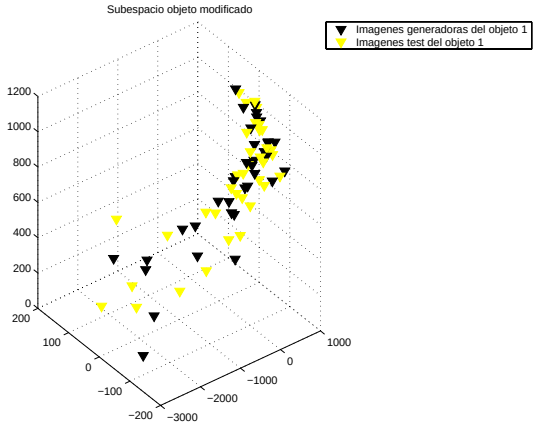


(b)

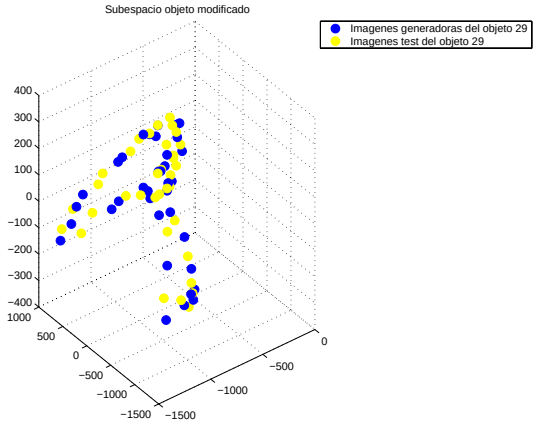


(c)

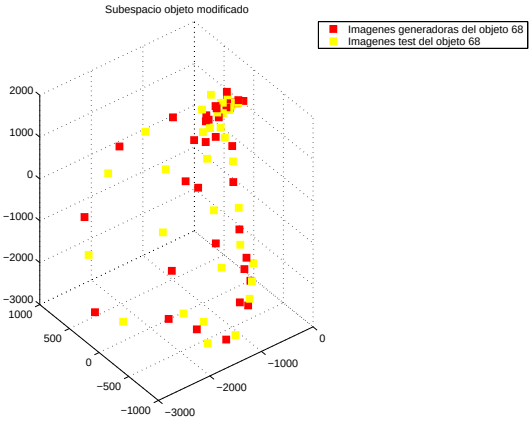
Figura 4.9: Proyección de las imágenes de test del objeto 68 sobre cada uno de los subespacios objeto.



(a)

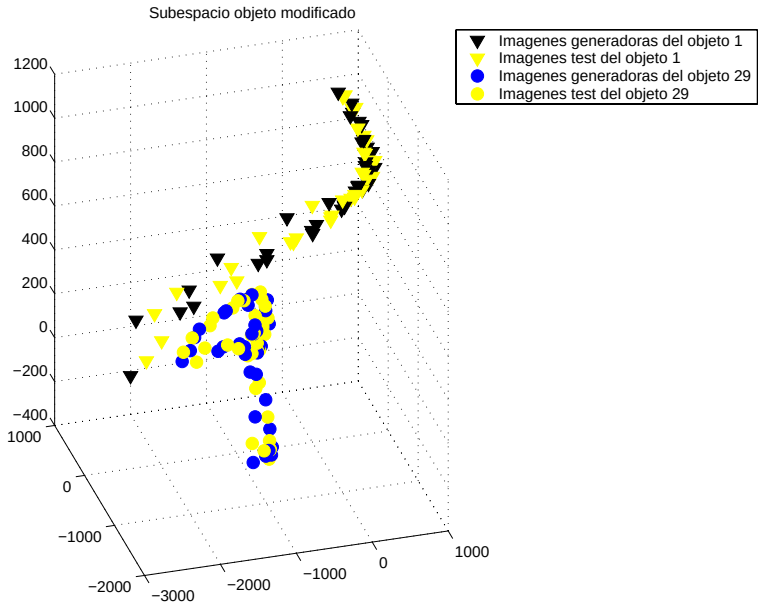


(b)

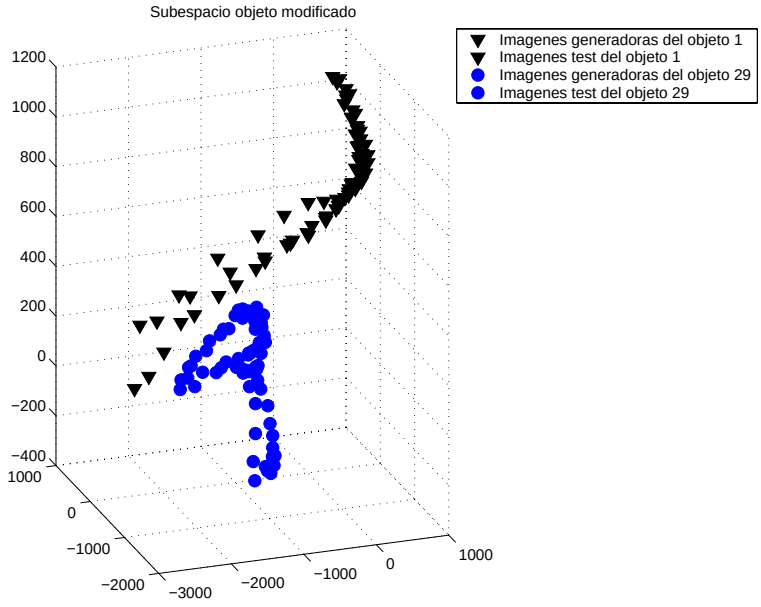


(c)

Figura 4.10: Subespacios objeto modificados



(a)



(b)

Figura 4.11: Subespacios objeto modificados

Es evidente que la ecuación 4.7 equivale a la concatenación de las proyecciones en cada uno de los subespacios objeto:

$$\mathbf{P}_{om} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_t \\ \vdots \\ \mathbf{P}_T \end{pmatrix} \tag{4.8}$$

Aparentemente, la arquitectura propuesta produce una menor compresión de información que la basada en el subespacio universal, dado que el número de componentes de cada patron es mayor: aumenta linealmente con el número de objetos, T:

$$\mathbf{P}_t \in \Re^{K \times N} \rightarrow \mathbf{P}_{om} \in \Re^{K \cdot T \times N} \tag{4.9}$$

Sin embargo, el rango de valores que puede tomar el parámetro K es distinto en cada caso. K representa el número de componentes utilizadas en la representación y tiene como límite superior el número de ejemplos disponibles para generar el subespacio. En el primero de los casos (arquitectuta de subespacio universal), este límite es N (número de ejemplos de entrenamiento); pero en el segundo caso (arquitectuta de subespacio objeto modificado), este límite es igual al número de ejemplos de cada clase (igual a $\frac{N}{T}$, supuesto que las clases están uniformemente distribuidas).

Por lo tanto, si se elige el máximo número de componentes posible, se obtendría una compresión igual en ambos casos, tal y como muestra la siguiente expresión:

$$\mathbf{P}_t \in \Re^{K_1 \times N} = \Re^{N \times N} \rightarrow \mathbf{P}_{om} \in \Re^{K_2 \cdot T \times N} = \Re^{\frac{N}{T} \cdot T \times N} = \Re^{N \times N} \tag{4.10}$$

En el caso que no se elija el máximo valor posible de K (o, lo que es lo mismo, en el caso en que se descarten las componentes de menor varianza) las compresiones de dichas arquitecturas pueden ser diferentes, pudiéndose alcanzar una compresión menor si se utiliza la arquitectura de subespacio universal.

La arquitectura de subespacio universal es un método no supervisado, pues los vectores del subespacio se obtienen a partir de la matriz de datos sin tener en cuenta las clases de patrones distintas. Recordemos que en caso de utilizar PCA, estos vectores son las direcciones ortogonales de máxima varianza en los datos.

La arquitectura de subespacio objeto modificado es al contrario un método supervisado, pues la matriz de transformación se obtiene a partir de los vectores de cada uno de los subespacios objeto de cada una de las clases.

Al tratarse de un método supervisado, si los datos de entrenamiento son suficientemente representativos de las clases [111], es esperable un mejor comportamiento en cuanto a la capacidad de separación entre clases y, por tanto, en cuanto al porcentaje de clasificaciones correctas sobre los ejemplos de test. Estos resultados serán corroborados experimentalmente en el apartado siguiente, donde se confirma la superioridad de procedimiento de extracción de características propuesto en la presente tesis (subespacio objeto modificado) respecto de los procedimientos existentes (subespacio universal).

Cabe plantearse si esta mejor capacidad de separación entre clases conlleva un incremento del coste computacional. Para realizar la comparación, se supondrá que el procedimiento de extracción de características utilizado en ambos casos es PCA[2].

Se comenzará calculando el coste computacional asociado a PCA sobre un subespacio universal. Si el tamaño de las imágenes de entrenamiento (número de píxeles) se define como M y el número total de ejemplos de entrenamiento como N, entonces el orden del coste computacional asociado a la generación de las componentes principales se puede expresar de acuerdo con la siguiente ecuación:

$$Coste_{gen.\ sub.\ univ.} = O(M^2N) + O(M^3) \tag{4.11}$$

donde el primero de los términos corresponde al coste necesario para el cálculo de la matriz de covarianzas (ecuación 3.20) y el segundo de los términos corresponde al coste asociado a la descomposición SVD (obtención de los valores y vectores propios). En el caso de emplear el *truco de Turk y Pentland* el coste computacional se reducirá a la siguiente expresión:

$$Coste_{gen.\ sub.\ univ.} = O(N^2M) + O(N^3) \tag{4.12}$$

puesto que en este caso la matriz de covarianzas utilizada para la descomposicón SVD es la referida en la ecuación 3.24, de tamaño $N \times N$.

Una vez calculado el subespacio, es necesario proyectar sobre él la imagen de cada nuevo objeto que se desea reconocer. Si asumimos que para la proyeción se utilizan K componentes de mayor varianza (donde $K \leq N$, necesariamente), entonces el coste computacional se puede representar como:

$$Coste_{proy.\ sub.\ univ.} = O(MK) \tag{4.13}$$

dado que la proyección equivale a la multiplicación de una matriz de dimension $K \times M$ (K componentes del subespacio) por una matriz de dimensión $M \times 1$ (imagen a clasificar de M píxeles).

[2]Esta suposición, como se verá más adelante, no supone pérdida de generalidad.

El coste total será pues el representado como

$$Coste_{sub.\ univ.} = O(N^2M) + O(N^3) + O(MK) \tag{4.14}$$

pero, dado que K está limitado superiormente por N ($K \leq N$), el último término de la expresión 4.14 es siempre menor o igual que $O(NM)$ y por lo tanto es despreciable respecto de $O(N^2M)$, con lo que el coste computacional total queda reducido a:

$$Coste_{sub.\ univ.} = O(N^2M) + O(N^3) \tag{4.15}$$

A continuación realizaremos el mismo análisis para PCA sobre una arquitectura subespacio objeto modificado. En este caso, supondremos que existen T clases distintas y que el número de ejemplos correspondiente a cada clase es el mismo, esto es, $\frac{N}{T}$. Por cada clase, es necesario obtener las componentes principales, es decir, obtener su subespacio objeto; el coste de obtener estas componentes principales para una sola clase será:

$$Coste_{gen.\ sub.\ obj.} = O(\left(\frac{N}{T}\right)^2 M) + O(\left(\frac{N}{T}\right)^3) \tag{4.16}$$

por lo tanto, repetir T veces el proceso anterior equivale a un coste:

$$Coste_{gen.\ sub.\ obj.\ mod.} = O(\frac{N^2M}{T}) + O(\frac{N^3}{T^2}) \tag{4.17}$$

Del mismo modo que en el caso anterior, una vez calculados los subespacios será necesario proyectar sobre ellos la imagen de cada nuevo objeto que se quiera reconocer. El número de componentes utilizado para la proyección será K, como en el caso anterior, y en esta ocasión se encontrará limitado por el número de ejemplos de entrenamiento de cada clase, esto es, $K \leq \frac{N}{T}$. El coste de la proyección sobre un subespacio objeto modificado para una sola clase será:

$$Coste_{proy.\ sub.\ obj.} = O(MK) \leq O(M\frac{N}{T}) \tag{4.18}$$

De este modo, el coste total para el total de los objetos en la arquitectura subespacio objeto modificado resultará:

$$Coste_{sub.\ obj.\ mod.} = O(\frac{N^2M}{T}) + O(\frac{N^3}{T^2}) + O(NM) \tag{4.19}$$

En la expresión anterior, dado que necesariamente se ha de cumplir que $T \leq N$, puesto que al menos hay un ejemplo de entrenamiento por cada clase, el tercer término

es despreciable frente al resto. Por lo tanto el coste computacional global asociado a la arquitectura subespacio objeto modificado resulta:

$$Coste_{sub.\ obj.\ mod.} = O(\frac{N^2 M}{T}) + O(\frac{N^3}{T^2}) \tag{4.20}$$

En conclusión, el subespacio objeto modificado propuesto en la presente tesis no sólo permite una mejor separación entre clases, sino que además supone un ahorro en coste computacional con respecto de la utilización del subespacio universal.

Un razonamiento similar se podría hacer para otros métodos de extracción de características distintos de PCA. En general, y a la vista del razonamiento realizado, se puede afirmar que cualquier método de extracción de características que tenga un coste computacional de grado uno (lineal) o superior (cuadrático, cúbico, ..) con el número de ejemplos de entrenamiento verificará:

$$Coste_{sub.\ obj.\ mod.} \leq Coste_{sub.\ obj.\ univ.} \tag{4.21}$$

la igualdad se cumplirá estrictamente cuando el coste sea exactamente lineal[3].

[3]Tanto PCA como ICA, LDA o NMF verifican la desigualdad anterior; únicamente RP, dado que presenta un coste computacional independiente del número de ejemplos de entrenamiento, deja de verificar dicha desigualdad.

4.3. Estudio del funcionamiento de la arquitectura propuesta con la librería COIL-100

En este apartado se presentan los resultados de reconocimiento obtenidos con la arquitectura propuesta en el apartado anterior (*subespacio objeto modificado*), con las imágenes de los objetos la librería COIL-20. El objetivo es el estudio del funcionamiento de la arquitectura propuesta con la técnica de PCA utilizando distancias métricas como clasificadores. Para ello se realiza una comparativa entre el funcionamiento de las arquitecturas subespacio universal y subespacio objeto modificado. Algunos de los experimentos que se muestran en este apartado se pueden consultar también en [170] y [172].

La librería COIL-20 [121] está formada por 1440 imágenes de 20 objetos diferentes (72 imágenes por cada objeto). Las imágenes se han adquirido de manera que entre dos imágenes consecutivas hay una rotación de 5º del objeto. En la figura 4.12 aparecen algunas imágenes pertenecientes a dicha librería.

Para realizar el estudio del funcionamiento de PCA en ambas arquitecturas y utilizando clasificadores basados en distancias métricas se han empleado tres canales de características representativos del objeto: el canal de apariencia en niveles de gris (canal I), la imagen de bordes del objeto (utilizando el detector de bordes Canny [31]) y la imagen del objeto binarizada.

4.3.1. Resultados con el subespacio universal PCA

Tal como se ha explicado anteriormente, en este método se utilizan imágenes de objetos diferentes para generar la matriz de datos y posteriormente el subespacio. En primer lugar se utilizará sólo una imagen de entrenamiento por cada objeto ($N = 1$, según la ecuación 4.1). Esta imagen de entrenamiento puede ser la más representativa del objeto, que en general es su vista frontal, o bien una imagen elegida aleatoriamente, al primer caso se le denominará *entrenamiento frontal* y al segundo *entrenamiento aleatorio*. El resto de las imágenes (71) se utilizarán para test.

En estas condiciones, donde el sistema de reconocimiento está formado por los objetos de la COIL-20, el tamaño de la matriz de covarianzas traspuesta $\mathbf{C_x}'$ es de 20x20, puesto que existen 20 objetos distintos, y por tanto se obtienen 20 vectores propios no-nulos como máximo. El mayor porcentaje de aciertos en la identificación de las imágenes test se obtiene utilizando todos los vectores propios puesto que así no se pierde información [90].

La tabla 4.1 muestra los resultados globales obtenidos con este algoritmo utilizando distancias métricas como clasificadores, en concreto, la distancia euclídea o norma L2 y la distancia de Mahalanobis:

$$L2 = \sum_{i=1}^{M} (x_i - y_i)^2 \tag{4.22}$$

$$MAH = \sum_{i=1}^{M} \frac{1}{\lambda_i} (x_i - y_i)^2 \tag{4.23}$$

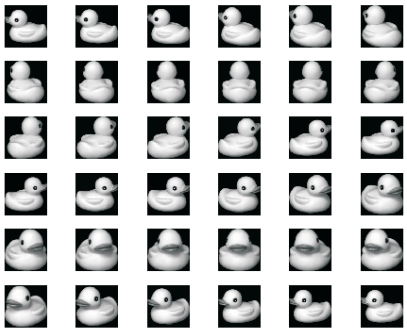
Los resultados señalan a la distancia de Mahalanobis como la medida de distancia que proporciona un mayor porcentaje de aciertos, lo cual es coherente con los principios estadísticos de PCA [48] [90] y con las comparativas de distancias métricas de otros autores como Baek [7] y Perlibakas [136]. Se aprecia también que se alcanzan los porcentajes más altos con las imágenes binarias y en niveles de gris ($64\,\%$).

La diferencia de funcionamiento entre canales y por objeto, utilizando entrenamiento frontal, se muestra con más detalle en las figuras 4.13 y 4.14. En la figura 4.13 se presentan los resultados obtenidos con la distancia de Mahalanobis y en la 4.14 los referentes a la norma L2. En ambos casos existen unos objetos que siempre son reconocidos (100 %), al menos, con el canal I, y son aquellos que se pueden considerar simétricos, es decir, que todas sus vistas son muy similares entre sí.

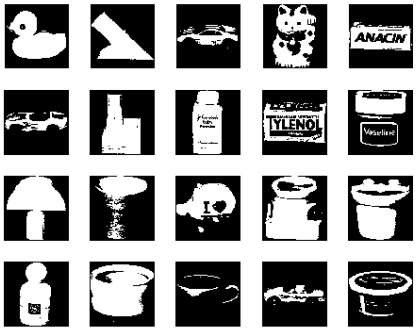
Los gráficos de la figuras 4.15, 4.16 y 4.17 muestran las tasas de reconocimiento por cada objeto con los tres canales, las dos distancias, y los dos modos de entrenamiento (frontal *vs.* aleatorio). En cada gráfico aparecen, por objeto, cuatro tasas de éxito asociadas a cuatro diferentes entrenamientos aleatorios y la media del conjunto de los experimentos aleatorios (un total de 25) en color negro, y el entrenamiento frontal, en color rojo. Las líneas discontínuas son la media global, es decir, la tasa de éxito en la identificación para el conjunto de todos los objetos (Tabla 4.1).

Globalmente, el entrenamiento frontal produce una tasa mayor que el aleatorio, salvo para algunos objetos en ciertos casos (p.ej. el objeto 10 con canal I y norma L2). El canal I y binario son los de las tasas mayores, aunque los objetos simétricos se reconocen al 100 % en cualquiera de los modos.

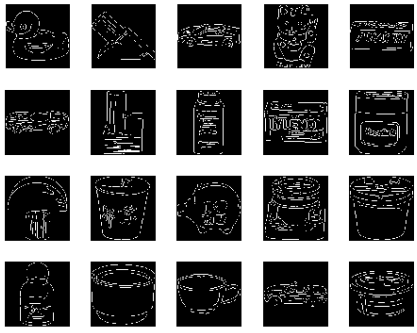
(a) Vista principal de los objetos de la COIL-20 en niveles de gris (canal I)



(b) 36 vistas del objeto 1 de la COIL-20



(c) Vista principal de los objetos de la COIL-20 en binario



(d) Vista principal de los objetos de la COIL-20 en imágenes de bordes

Figura 4.12: Librería COIL-20

Canal de características	I (niveles de gris)			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	57 %	52 %	64 %	60 %
Canal de características	Binario			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	56 %	55 %	64 %	61 %
Canal de características	Bordes (Canny)			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	23 %	20 %	52 %	53 %

Tabla 4.1: Tasas de éxito en el reconocimiento para los objetos de la librería COIL-20 utilizando subespacio universal PCA en los canales: niveles de gris, bordes y binario

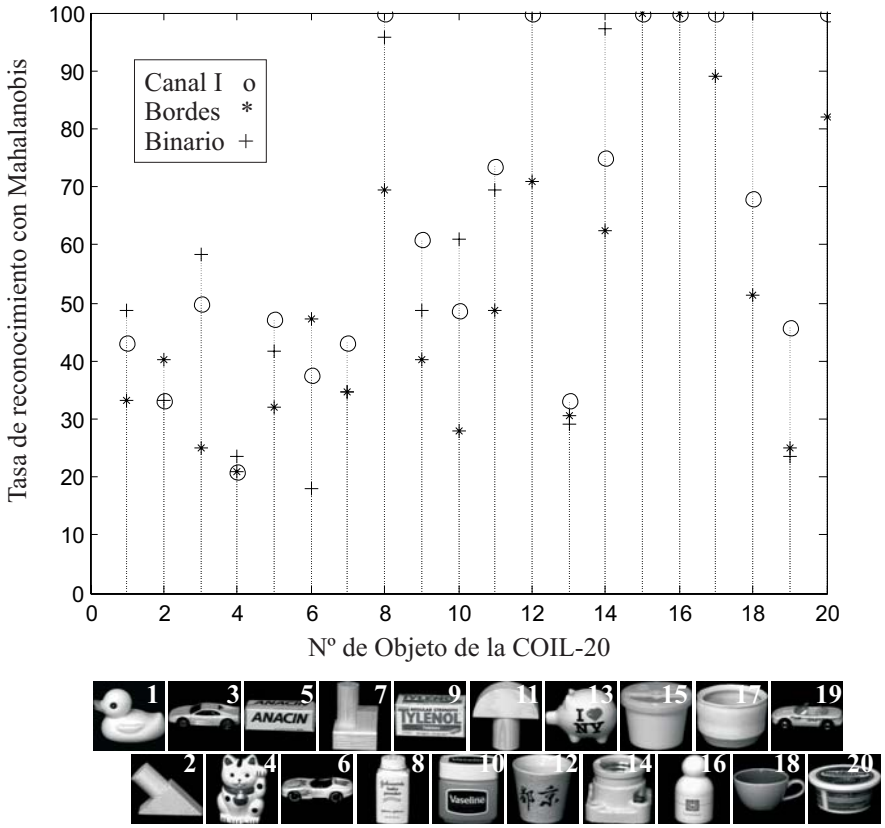


Figura 4.13: Tasa de éxito en el reconocimiento con subespacio universal PCA para las imagenes de la COIL-20 en los tres canales, con distancia de Mahalanobis y entrenamiento tipo *frontal*. Los objetos considerados simétricos tienen una tasa del 100 %

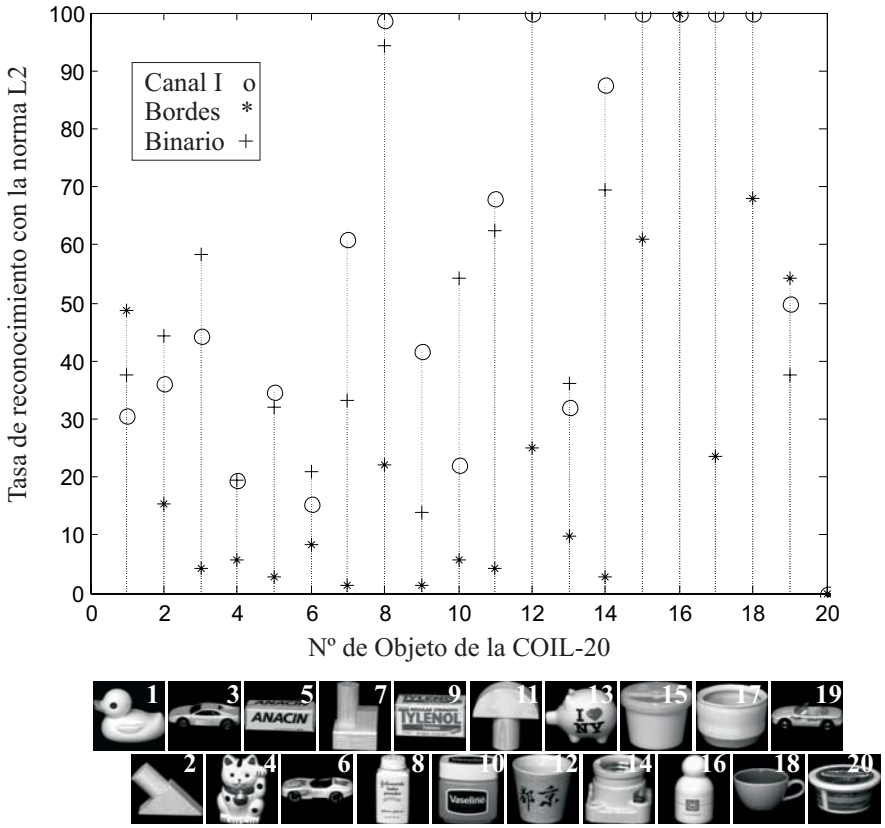
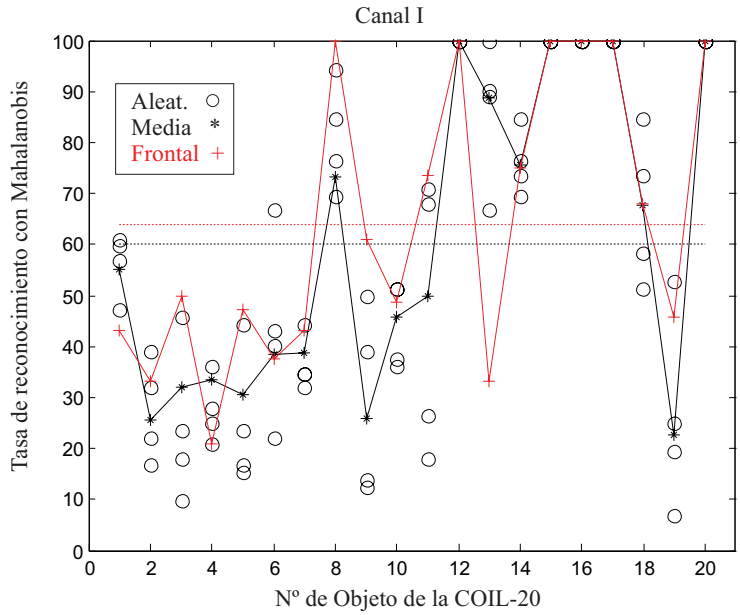
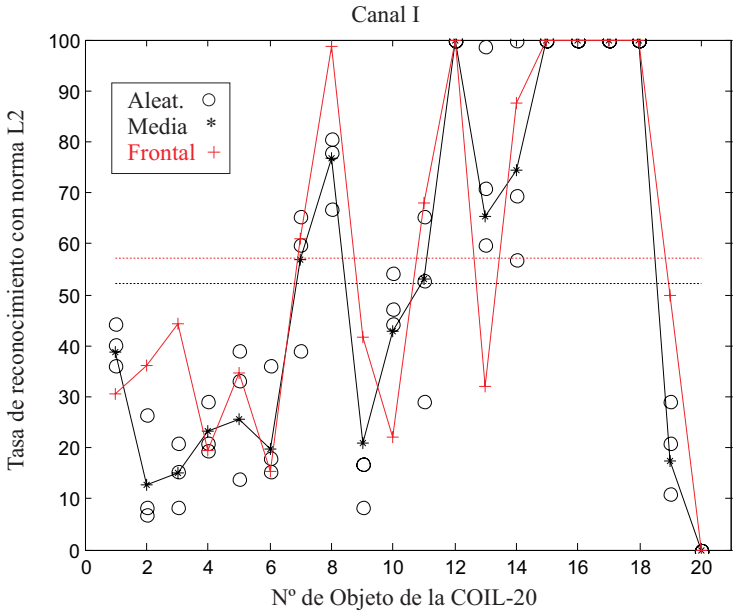


Figura 4.14: Tasa de éxito en el reconocimiento con subespacio universal PCA para las imagenes de la COIL-20 en los tres canales, con la norma L2 y entrenamiento tipo *frontal*. Los objetos considerados simétricos tienen una tasa del 100 %

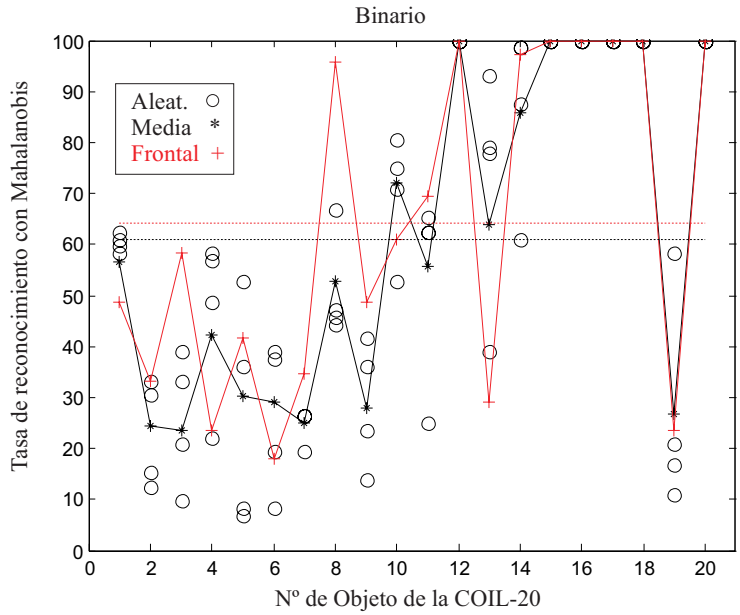


(a) canal I, MAH

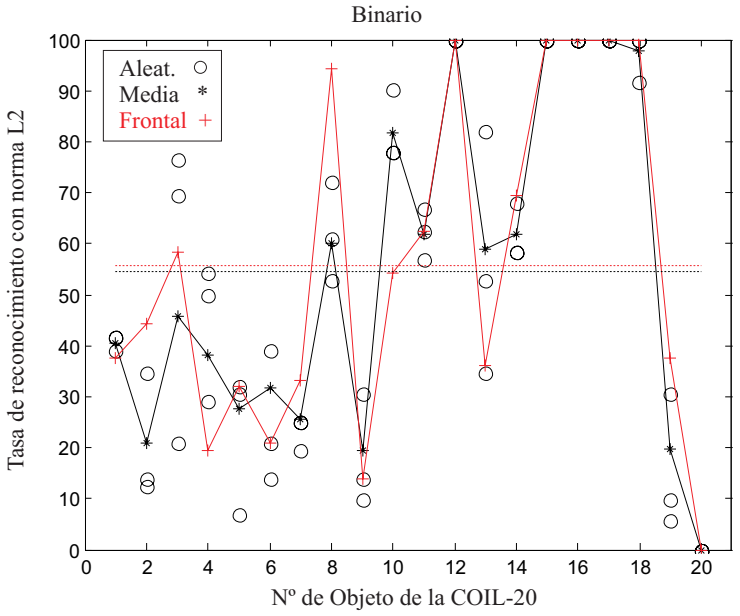


(b) canal I, norma L2

Figura 4.15: Entrenamiento *frontal* vs *aleatorio* para el canal I

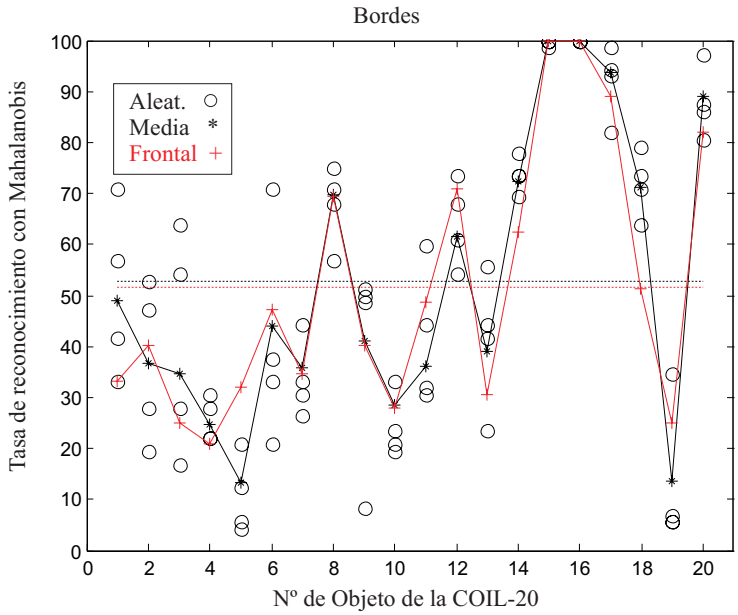


(a) Binario, MAH

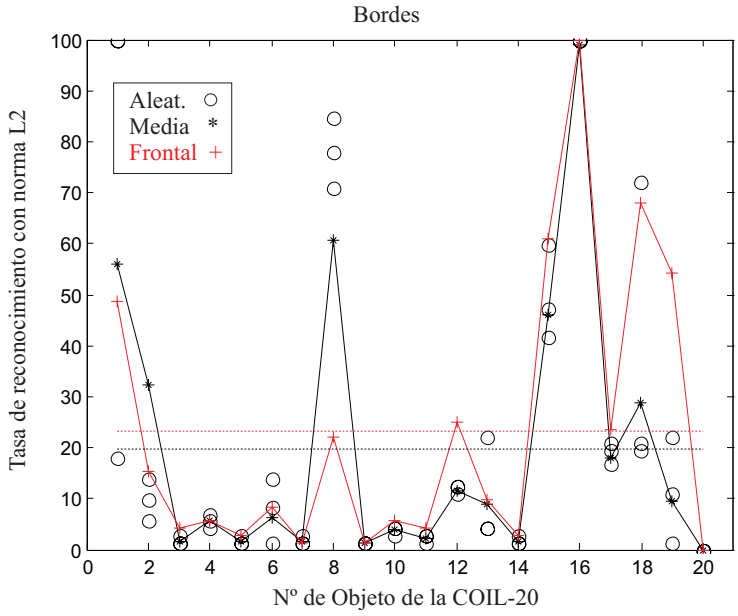


(b) Binario, norma L2

Figura 4.16: Entrenamiento *frontal* vs *aleatorio* para el canal binario



(a) Bordes, MAH



(b) Bordes, norma L2

Figura 4.17: Entrenamiento *frontal* vs *aleatorio* para el canal de bordes

También se han realizado pruebas de la robustez del funcionamiento del método PCA con tipo de subespacio universal ante oclusiones parciales de los objetos bajo las mismas condiciones de entrenamiento y test ($N = 1$ por objeto para el entrenamiento, y $N = 71$ por objeto para test). Para la realización de estos experimentos se generaron oclusiones de los objetos originales mediante la eliminación de un 25 % y un 50 % de la imagen del objeto original (ver figura 4.18). La tabla 4.2 muestra un resumen de los resultados obtenidos con las distancias métricas para los experimentos con oclusiones: de nuevo, la distancia de Mahalanobis proporciona una tasa mayor. El gráfico de la figura 4.19 muestra los porcentajes de reconocimiento por objeto para el canal I en el caso de un 25 % de oclusión. La mayoría de los objetos se confunden (<50 %) salvo los simétricos (100 %).

(a) Imágenes de la COIL20 con una oclusion del 25 %

(b) Imágenes de la COIL20 con una oclusion del 50 %

Figura 4.18: Oclusiones

Canal de características	I (niveles de gris)			
Tipo Oclusion	25 %		50 %	
Norma	L2	MAH	L2	MAH
Tasa reconocimiento	9 %	51 %	7 %	16 %
Canal de características	Binario			
Tipo Oclusion	25 %		50 %	
Norma	L2	MAH	L2	MAH
Tasa reconocimiento	7 %	43 %	5 %	15 %
Canal de características	Bordes (Canny)			
Tipo Oclusion	25 %		50 %	
Norma	L2	MAH	L2	MAH
Tasa reconocimiento	6 %	39 %	11 %	32 %

Tabla 4.2: Tasas de éxito en el reconocimiento con oclusiones

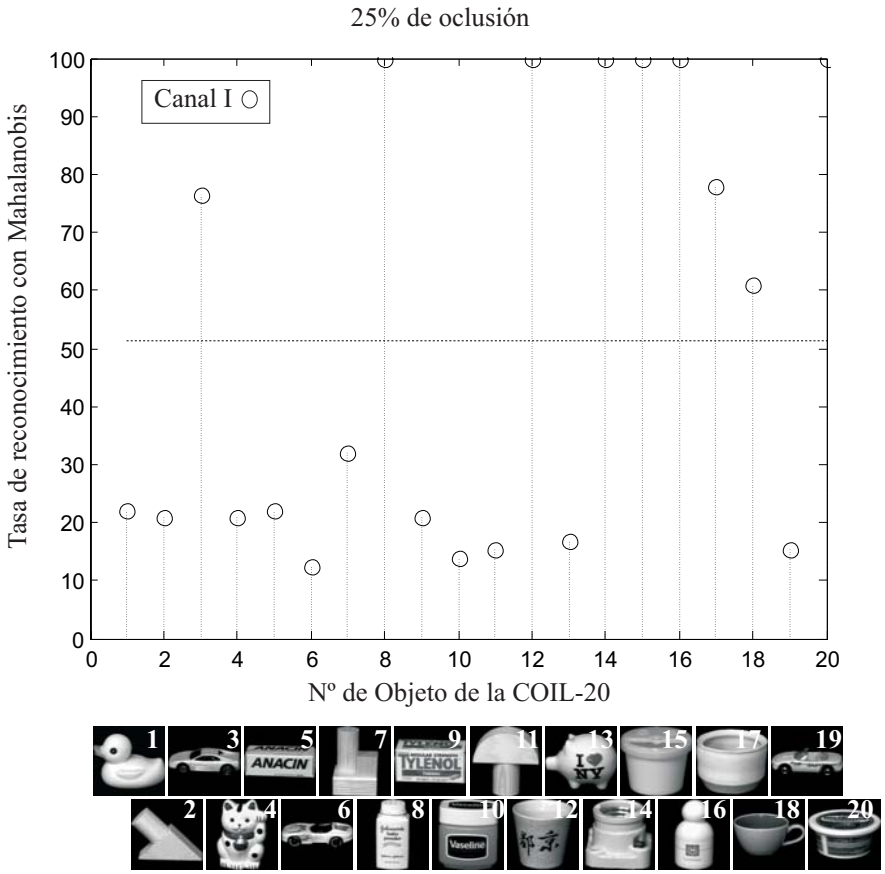


Figura 4.19: Tasa de éxito en el reconocimiento para canal I con oclusion del 25 %, los objetos simétricos se reconocen al 100 %

4.3.2. Uso de más vistas para subespacio universal

En la sección anterior se mostraron las tasas de éxito en el reconocimiento obtenidas con la arquitectura universal utilizando una sola imagen por objeto para obtener el subespacio PCA. A continuación se presentan los resultados experimentales obtenidos con una arquitectura tipo subespacio universal pero utilizando más imágenes de entrenamiento por objeto para obtener los vectores de la transformación PCA. La tabla 4.3 muestra los resultados obtenidos con la norma L2, y la tabla 4.4 muestra los porcentajes calculados con la distancia de Mahalanobis. Como era de esperar son superiores al caso anterior de entrenamiento con sólo una imagen por objeto, y la tasa de éxito en el reconocimiento aumenta con el número de vistas utilizadas para generar el subespacio.

	Norma L2, tasa de éxito (%)		
	Canal de características		
# Vistas	I (gris)	Binario	Bordes
5	59	48	27
10	62	43	30
15	64	45	34
20	70	57	39
25	73	57	43
30	77	58	44
35	79	63	59

Tabla 4.3: Resultados experimentales de reconocimiento con distancia euclídea para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la COIL-20.

	MAH, tasa de éxito (%)		
	Canal de características		
# Vistas	I (gris)	Binario	Bordes
5	68	68	47
10	69	63	50
15	70	65	54
20	73	67	59
25	74	67	64
30	76	65	62
35	76	63	63

Tabla 4.4: Resultados experimentales de reconocimiento con distancia de Mahalanobis para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la COIL-20.

Asímismo se han realizado pruebas de la robustez del funcionamiento del método PCA con tipo de subespacio universal de varias vistas ante oclusiones parciales de los objetos (de un 25 % y un 50 % de la imagen del objeto original). La tabla 4.5 muestra un resumen de los resultados obtenidos con las distancias métricas para los experimentos con oclusiones utilizando 30 vistas de cada objeto para la generación del subespacio. Como en el caso equivalente anterior, la distancia de Mahalanobis proporciona una tasa mayor que la distancia euclídea, y en conjunto son superiores al caso de entrenamiento $N = 1$.

Canal de características	I (niveles de gris)			
Tipo Oclusion	25 %		50 %	
Norma	L2	MAH	L2	MAH
Tasa reconocimiento	35 %	68 %	14 %	43 %
Canal de características	Binario			
Tipo Oclusion	25 %		50 %	
Norma	L2	MAH	L2	MAH
Tasa reconocimiento	22 %	54 %	18 %	35 %
Canal de características	Bordes (Canny)			
Tipo Oclusion	25 %		50 %	
Norma	L2	MAH	L2	MAH
Tasa reconocimiento	19 %	48 %	33 %	60 %

Tabla 4.5: Tasas de éxito en el reconocimiento con oclusiones para la arquitectura subespacio universal utilizando 30 vistas para el entrenamiento.

4.3.3. Resultados con los subespacios objeto modificados PCA

En este tipo de arquitectura se genera un subespacio por cada objeto y se utilizan la unión de todos para realizar la proyección de cada imagen (según las ecuaciones 4.6 y 4.7). En función del conjunto de imágenes elegidas para el entrenamiento o la generación del subespacio se distinguirán dos tipos de entrenamiento: el denominado *completo*, en el que se utilizan 36 imágenes intercaladas, es decir, entre cada imagen hay una rotación de 10º, por lo que el conjunto de entrenamiento recoge todas las vistas del objeto; y el entrenamiento *aleatorio*, donde se seleccionan 36 imágenes al azar.

Como no todos los objetos presentan el mismo grado de dificultad en el proceso de reconocimiento, se han separado en tres grupos según la variabilidad de sus vistas [4]:

- conjunto I: 7,12, 15,16,17,18,20
- conjunto II: 1,4, 8, 10, 11,13
- conjunto III:2,3,5,6,9,14,19

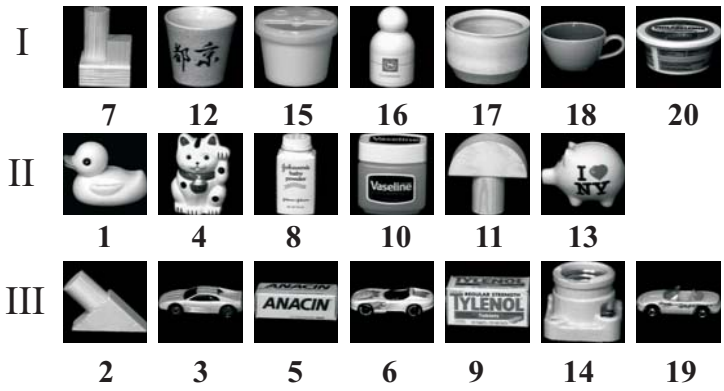


Figura 4.20: Objetos de la COIL-20 en grupos según la variablidad de sus vistas

El conjunto I engloba a los objetos más simétricos de toda la librería, es decir, que todas sus vistas son muy similares entre sí. El conjunto II reúne a los objetos que presentan alguna diferencia considerable entre sus vistas o poses y por último, el conjunto III agrupa a los objetos que presentan vistas muy diferentes entre sí.

[4]los objetos se han numerado por filas, empezando por el pato de goma (1)

En la tabla 4.6 se presentan los resultados obtenidos para el canal de características I, por conjuntos, variando el tamaño del subespacio objeto (número de vectores)[5], y utilizando dos tipos de medidas de distancias con el entrenamiento tipo *completo* para las imágenes originales. En este experimento las normas L2 y L1 proporcionan los mejores porcentajes de aciertos (85 % para la L2), mientras que la distancia de Mahalanobis es la que peor funciona (19 %) a diferencia de lo que ocurría con el subespacio universal. El funcionamiento de la distancia de Mahalanobis con las imágenes de los objetos proyectadas en los subespacios objeto se muestra en la figura 4.21: los objetos simétricos se confunden totalmente con el resto. Ocurre todo lo contrario que sucedía con los subespacios universales, que proporcionaban los porcentajes de éxito más elevados para los objetos simétricos utilizando la distancia de Mahalanobis.

Básicamente, los vectores que definen la transformación PCA son los vectores propios de la matriz de covarianzas de los datos en orden decreciente a sus correspodientes valores propios. Siendo cada valor propio igual a la varianza de los datos en la dirección correspondiente de su vector propio. Si las distancias entre las proyecciones (los puntos residentes en los subespacios PCA) son medidas con la distancia de Mahalanobis, se está ponderando cada componente por su valor propio (la varianza en esa dirección). Dicho de forma más sencilla, se están escalando las componentes. Si no se realiza este escalado, es decir, si se utiliza la norma L2 como medida de distancia, todas las componentes tendrán el mismo peso. Utilizar la distancia de Mahalanobis es equivalente a utilizar la transformación de $whitening$ o blanqueo, que se detallará más adelante en el próximo capítulo, junto con la norma L2 como medida de distancia.

El utilizar la distancia de Mahalanobis puede ser beneficioso si se utiliza la transformación de PCA estricta [90], es decir, se utilizan subespacios universales (u objeto), pero no una concatenación de éstos. En el gráfico de la figura 4.21 los objetos que se confunden totalmente son los mismos que con el subespacio universal y la distancia de Mahalanobis se identificaban siempre correctamente (figura 4.13). Sin embargo, los objetos que presentan mayor variabilidad en sus vistas tienen tasas mayores con la norma L2 utilizando subespacios objeto como se muestra en el gráfico de la figura 4.22.

Por tanto, los resultados experimentales señalan que no resulta interesante utilizar la distancia de Mahalanobis si se utilizan subespacios objeto modificados. Sin embargo, la distancia euclídea sí proporciona unas tasas de acierto muy elevadas.

[5]El tamaño del subespacio objeto modificado es igual al número de vectores de cada subespacio multiplicado por el número de objetos del sistema.

Canal de características	I (niveles de gris)						
# Vectores	5	10	15	20	25	30	35
Norma L2	tasa de éxito (%)						
I	92	98	99	99	100	100	100
II	60	89	85	89	92	92	94
III	42	55	63	62	61	60	61
total	65	81	82	83	84	84	85
Norma L1	tasa de éxito (%)						
I	93	96	98	98	98	99	100
II	58	83	80	84	87	90	92
III	39	53	62	62	56	51	54
total	63	77	80	81	80	80	82

Tabla 4.6: Resultados experimentales de reconocimiento con la arquitectura subespacio objeto modificado PCA para las imágenes de la COIL-20 en el canal I(niveles de gris) con el entrenamiento tipo *completo*

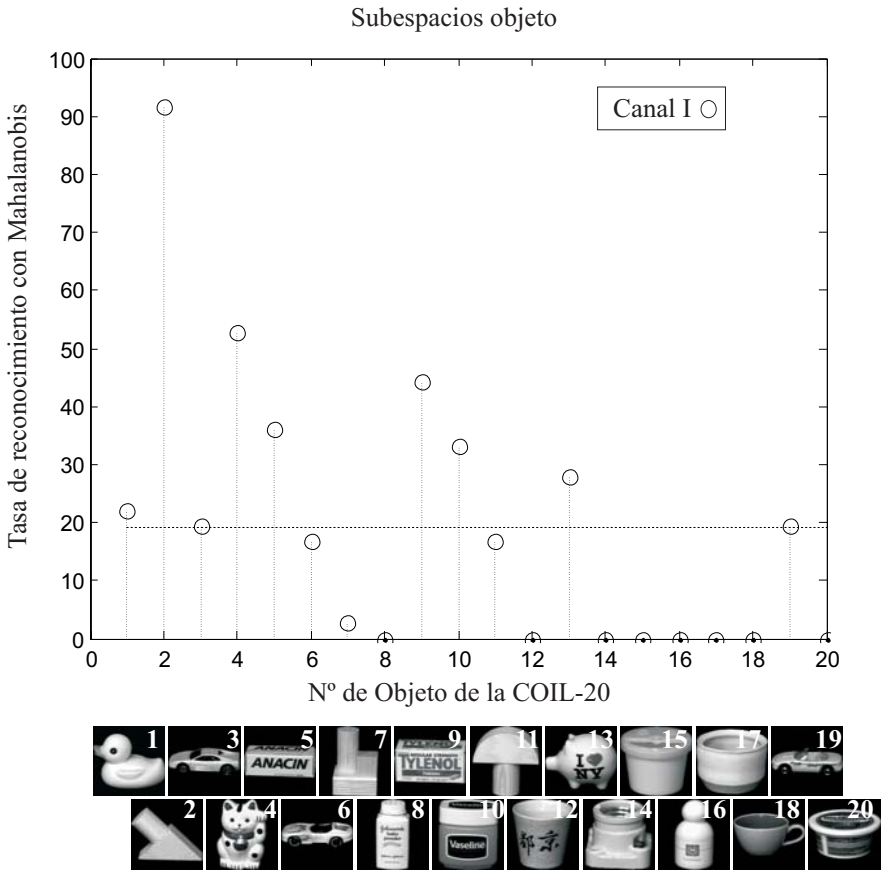


Figura 4.21: Tasa de éxito en el reconocimiento para canal I, con la arquitectura subespacio objeto modificado PCA y distancia de Mahalanobis (36 vistas de entrenamiento)

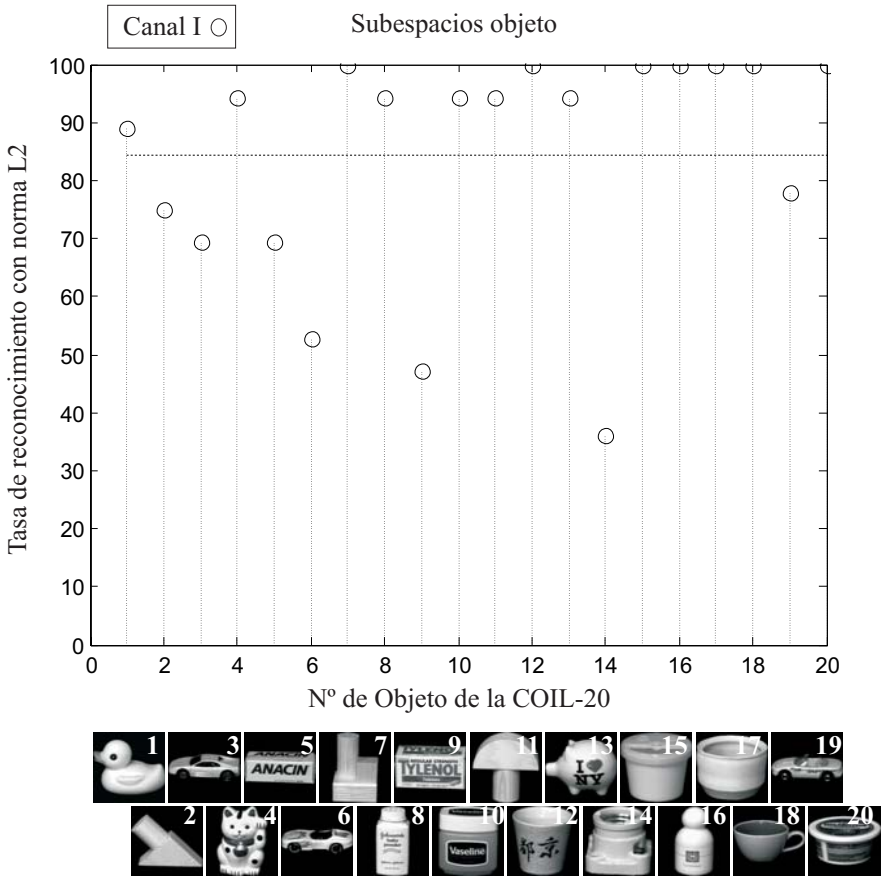


Figura 4.22: Tasa de éxito en el reconocimiento para canal I, con la arquitectura subespacio objeto modificado PCA y norma L2 (36 vistas de entrenamiento)

Canal de características	I (niveles de gris)						
# Vectores	5	10	15	20	25	30	35
Norma L2	tasa de éxito (%)						
I	85	89	92	96	96	96	97
II	57	72	72	79	81	84	88
III	38	53	54	54	55	57	58
total	60	71	73	76	77	79	81

Tabla 4.7: Resultados experimentales de reconocimiento con subespacios objeto PCA para las imagenes de la COIL-20 en el canal I(niveles de gris) con el entrenamiento tipo *aleatorio*

Canal de características	Binario						
# Vectores	5	10	15	20	25	30	35
Norma L2	tasa de éxito (%)						
I	80	84	78	77	76	74	68
II	81	79	71	65	58	54	50
III	53	54	52	44	39	36	31
total	71	72	67	62	58	55	50

Tabla 4.8: Resultados experimentales de reconocimiento con la arquitectura subespacio objeto modificado PCA para las imagenes de la COIL-20 en el canal Binario y con el entrenamiento tipo *completo*.

Canal de características	Bordes				
# Vectores	5	10	20	33	35
Norma L2	tasa de éxito (%)				
I	90	95	98	100	100
II	67	83	95	99	99
III	50	54	74	87	87
total	69	77	89	95	95

Tabla 4.9: Resultados experimentales de reconocimiento con la arquitectura subespacio objeto modificado PCA para las imagenes de la COIL-20 en el canal Bordes y con el entrenamiento tipo *completo*.

La Tabla 4.7 muestra el mismo experimento que la Tabla 4.6 pero utilizando el entrenamiento tipo aleatorio: se obtiene un porcentaje algo menor que el mejor entrenamiento pero más alto de lo esperado (81 %). En ambas tablas se aprecia la separación entre grupos, el conjunto I es muy fácil de reconocer casi al 100 % utilizando apenas 5 vectores.

Realizando el mismo estudio pero con las imágenes binarias y de bordes de la librería COIL-20 se obtiene que el porcentaje de aciertos con las imágenes binarias para el mejor caso (mejor entrenamiento y norma L2) desciende hasta el 50 %, mientras que con las imágenes de bordes conseguimos un sorprendente 95 % de aciertos utilizando la distancia euclídea. Por tanto, si se realiza una detección de bordes en las imágenes de entrenamiento antes de realizar el cálculo del subespacio, la tasa de éxito en el reconocimiento aumenta considerablemente.

4.4. Estudio del funcionamiento de la arquitectura propuesta con la librería de objetos UMH-15

En este apartado se presentan los resultados de reconocimiento obtenidos con la arquitectura propuesta, con las imágenes de los objetos de la librería UMH-15. Como antes, el objetivo es el estudio del funcionamiento de la arquitectura propuesta con la técnica de PCA utilizando distancias métricas como clasificadores. Para ello se realiza una comparativa entre el funcionamiento de las arquitecturas subespacio universal y subespacio objeto modificado. Algunos resultados preliminares se pueden consultar en [171].

La librería UMH-15 ha sido generada expresamente para realizar este estudio y contrastar sus resultados con los obtenidos con la COIL-20. Está formada por 360 imágenes de 15 objetos diferentes (24 imágenes por cada objeto). Las imágenes se han adquirido de manera que entre dos imágenes consecutivas hay una rotación de 15º del objeto. En la figura 4.23 aparecen los 15 objetos pertenecientes a dicha librería y en la figura 4.24 se muestran todas las vistas de dos de sus objetos. Las imágenes de los objetos de la librería UMH-15 presentan un mayor número de sombras y los objetos no se encuentran tan uniformemente escalados como en la librería COIL-20, por lo que su reconocimiento resulta más complejo. Con ello se pretende validar el funcionamiento de las arquitecturas de reconocimiento testeadas en las secciones precedentes.

Figura 4.23: Librería UMH-15

(a) Objeto 8

(b) Objeto 10

Figura 4.24: Objetos 8 y 10 de la librería UMH-15

4.4.1. Resultados con el subespacio universal PCA

Como en el apartado equivalente con los objetos de la COIL-20, en esta sección se presentan los resultados de reconocimiento obtenidos utilizando la arquitectura con sólo una imagen de entrenamiento por cada objeto. Como antes, esta imagen de entrenamiento puede ser la más representativa del objeto (entrenamiento frontal) o bien una imagen al azar (entrenamiento aleatorio). El resto de las imágenes (23) se utilizan para test.

La tabla 4.10 muestra los resultados globales obtenidos utilizando la norma L2 y la distancia de Mahalanobis en los tres canales de características antes utilizados (niveles de gris, binario y bordes). Las tasas de reconocimiento alcanzadas en este caso son algo superiores a las de la librería COIL-20, lo que puede ser debido al menor número de objetos en la base de datos o bien a que proporcionan clases más separables (son más diferentes entre sí). Sin embargo, como antes, se aprecia que la distancia de Mahalanobis proporciona unos porcentajes más altos que la distancia euclídea.

Canal de características	I (niveles de gris)			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	68 %	69 %	76 %	75 %
Canal de características	Binario			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	60 %	65 %	71 %	69 %
Canal de características	Bordes (Canny)			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	34 %	31 %	61 %	60 %

Tabla 4.10: Tasas de éxito en el reconocimiento para los objetos de la librería UMH-15 utilizando subespacio universal PCA en los canales: niveles de gris, bordes y binario.

Para este conjunto de objetos, aquellos que presentan mayor semejanza entre sus vistas y que por tanto, se considerarán *simétricos* son los que se muestran en la figura 4.25. Para este grupo de objetos, los porcentajes de reconocimiento son más altos que el global de toda la librería que se muestra en la tabla 4.10. Estos porcentajes se presentan en la tabla 4.11.

Figura 4.25: Objetos simétricos de la librería UMH-15

Canal de características	I (niveles de gris)			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	97 %	94 %	100 %	100 %
Canal de características	Binario			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	89 %	88 %	94 %	94 %
Canal de características	Bordes (Canny)			
Entrenamiento	frontal	aleatorio	frontal	aleatorio
Norma	L2	L2	MAH	MAH
Tasa reconocimiento	84 %	78 %	96 %	90 %

Tabla 4.11: Tasas de éxito en el reconocimiento para los objetos simétricos de la librería UMH-15 utilizando subespacio universal PCA en los canales: niveles de gris, bordes y binario.

4.4.2. Uso de más vistas para subespacio universal

En la sección anterior se mostraron las tasas de éxito en el reconocimiento obtenidas con la arquitectura universal utilizando una sola imagen por objeto para obtener el subespacio PCA. A continuación se presentan los resultados experimentales obtenidos con una arquitectura tipo subespacio universal pero utilizando más imágenes de entrenamiento por objeto para obtener los vectores de la transformación PCA. La tabla 4.12 muestra los resultados obtenidos con la norma L2, y la tabla 4.13 muestra los porcentajes calculados con la distancia de Mahalanobis. Como era de esperar son superiores al caso anterior de entrenamiento con sólo una imagen por objeto, y la tasa de éxito en el reconocimiento aumenta con el número de vistas utilizadas para generar el subespacio.

	Norma L2, tasa de éxito (%)		
	Canal de características		
# Vistas	I (gris)	Binario	Bordes
2	74	62	36
4	77	64	38
8	79	67	39
12	82	71	42

Tabla 4.12: Resultados experimentales de reconocimiento con distancia euclídea para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la UMH-15.

	MAH, tasa de éxito (%)		
	Canal de características		
# Vistas	I (gris)	Binario	Bordes
2	84	78	67
4	89	83	71
8	92	83	72
12	97	85	76

Tabla 4.13: Resultados experimentales de reconocimiento con distancia de Mahalanobis para la arquitectura subespacio universal de varias vistas PCA con las imágenes de la UMH-15.

4.4.3. Resultados con los subespacios objeto modificados PCA

En las tablas 4.14 y 4.15 se presentan los resultados obtenidos para los tres canales de características y las dos distancias métricas, variando el tamaño del subespacio objeto. En la tabla, *# Vectores* indica el número de componentes de cada uno de los subespacios objeto; el subespacio objeto modificado tendrá $(15 \cdot \#\ \textit{Vectores})$ dimensiones. El entrenamiento es tipo *completo*, es decir, se utilizan 12 imágenes intercaladas (rotación de 30° entre cada par de imágenes) para el cómputo de los vectores y las otras 12 imágenes restantes se utilizan para test.

Como ocurría con los objetos de la COIL-20, en este experimento la norma L2 proporciona mejores porcentajes de aciertos que la distancia de Mahalanobis, a diferencia de lo que ocurría con el subespacio universal.

	MAH, tasa de éxito (%)		
	Canal de características		
# Vectores	I (gris)	Binario	Bordes
4	15	13	6
6	18	18	9
8	18	16	12
10	21	18	15
12	20	14	13

Tabla 4.14: Resultados experimentales de reconocimiento con distancia de Mahalanobis para la arquitectura subespacio objeto modificado PCA con las imágenes de la UMH-15.

	Norma L2, tasa de éxito (%)		
	Canal de características		
# Vectores	I (gris)	Binario	Bordes
1	60	52	66
2	64	55	68
3	68	56	66
4	68	60	72
5	71	64	79
6	75	64	71
7	79	70	75
8	83	68	88
9	85	74	94
10	96	79	98
11	92	74	90
12	90	71	94

Tabla 4.15: Resultados experimentales de reconocimiento con distancia euclídea para la arquitectura subespacio objeto modificado PCA con las imágenes de la UMH-15.

4.5. Conclusiones

A la vista de los resultados con los objetos de las librerías COIL-20 y UMH-15, la respuesta a la cuestión sobre qué arquitectura y distancia son las idóneas para la construcción del sistema de reconocimiento basado en PCA, dependerá de la variabilidad en las vistas de los objetos a reconocer. Si los objetos siempre muestran la misma cara al sistema o todas sus vistas son muy similares, es decir, los podemos considerar simétricos, la arquitectura universal con sólo una imagen generadora por objeto produce unas tasas de éxito en el reconocimiento aceptables, aún siendo el método más simple computacionalmente y que además produce la reducción de la dimensión de los patrones mayor, por lo que el clasificador será más rápido que con cualquier otro método.

Sin embargo, para objetos no simétricos, es necesario utilizar más imágenes por cada objeto durante la generación del subespacio universal o bien emplear la arquitectura propuesta en esta tesis.

En cuanto a la medida de distancia a utilizar, lo más eficiente es utilizar la distancia de Mahalanobis con subespacios universales y la norma L2 con los subespacios objeto modificados.

En resumen, los resultados experimentales presentados en este apartado muestran que la utilización de subespacios objeto modificados de acuerdo con la arquitectura propuesta resulta beneficiosa para aumentar la capacidad de reconocimiento de los objetos con más variabilidad en sus vistas, en comparación con la utilización de subespacios universales.

Capítulo 5

Métodos de extracción de características propuestos

En este capítulo se estudian analítica y experimentalmente los métodos de extracción de características expuestos en el capítulo 3 en el ámbito del reconocimiento visual de objetos. Básicamente se analizan los métodos de ICA, RP y NMFSC frente al método clásico de PCA.

En los últimos años han aparecido numerosos trabajos en los que se afirma la superioridad de la técnica de ICA frente al método tradicional de PCA, aunque también existen fuentes que afirman la idea opuesta. En este capítulo se incluye un estudio detallado de las similitudes y diferencias de ambos métodos y se aclara bajo qué circunstancias ICA puede superar los resultados de PCA.

Como aportación de la presente tesis, se estudia la aplicabilidad del método RP como extractor de características para reconocimiento de objetos. En este capítulo se muestran tanto las ventajas teóricas esperables como los resultados experimentales obtenidos, siempre comparados frente a PCA como método clásico de extracción de características.

Por último, se describen los resultados de un nuevo algoritmo de NMF bajo la arquitectura de subespacio objeto modificado. Siendo su principal ventaja, la capacidad para producir representaciones locales de los objetos, lo que permite mejorar el reconocimiento de los objetos en entornos no estructurados, es decir, en presencia de una gran cantidad de objetos, oclusiones o iluminación variable.

5.1. ICA vs. PCA

Las propuestas previas de sistemas de reconocimiento basados en apariencia se detallaron en el capítulo 2, donde se resaltó el hecho que recientemente existen numerosos estudios experimentales en los que se afirma la superioridad de ICA frente al método clásico de PCA como técnica de extracción de características en los sistemas de reconocimiento basados en la apariencia (ver Tabla 2.3). Una de las aportaciones de esta tesis es estudiar el porqué de éstos resultados contradictorios y aclarar bajo qué circunstancias ICA puede superar los resultados de PCA.

5.1.1. Capacidad de separación de clases de cada uno de los métodos

Para estudiar la capacidad de separación entre clases, es decir, entre objetos diferentes dentro del mismo sistema de reconocimiento, de cada uno de los dos métodos es necesario resaltar las sutiles diferencias que existen entre las técnicas de PCA, SVD[1], ICA y blanqueo de datos o *whitening*. Aunque todas estas técnicas están intrínsecamente muy relacionadas, *whitening* y PCA difieren en un aspecto importante que puede afectar considerablemente a los resultados del clasificador utilizado en el sistema de reconocimiento de objetos.

5.1.1.1. PCA, SVD y *whitening*

En la arquitectura de subespacio universal, la matriz de datos original $\mathbf{X}$, está compuesta por las imágenes de entrenamiento de los diferentes objetos del sistema, de manera que cada columna contiene los valores de los píxeles del vector imagen, $\mathbf{x}$.

Suponiendo que la matriz $\mathbf{X}$ está centrada (se le ha restado la imagen media a cada una de las columnas), la transformada PCA de la matriz $\mathbf{X}$ de tamaño $m \times n$ viene definida por

$$\mathbf{Y} = \mathbf{U}^T\mathbf{X} \tag{5.1}$$

donde $\mathbf{U}$ es una matriz ortonormal de tamaño $m \times m$. Como se explica en la sección 3.2.1, las componentes principales (las columnas de $\mathbf{U}$) son las direcciones de máxima varianza de los datos bajo la restricción de ortogonalidad mutua. Exactamente, los vectores de componentes principales son los vectores propios de la matriz de covarianzas de los datos $C_{\mathbf{X}} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$ en orden decreciente a sus correspondientes valores propios. El valor propio de mayor valor es igual al valor de la máxima varianza, mientras que su correspondiente vector propio determina la dirección dentro del espacio imagen con esa máxima varianza en los datos. Además la transformación definida en la ecuación 5.1 proporciona componentes no correladas.

[1] Singular Value Decomposition

En la ecuación 5.1, $\mathbf{Y}$ es la matriz original de datos proyectada sobre el subespacio PCA definido por los vectores propios. A partir de esa relación, es posible reducir la dimensión de los datos simplemente seleccionando un subconjunto de k vectores propios del conjunto total ($\Re^m \rightarrow \Re^k$),

$$\widetilde{\mathbf{Y}} = \widetilde{\mathbf{U}}^T\mathbf{X} \tag{5.2}$$

de esta forma $\widetilde{\mathbf{Y}}$ es una matriz de tamaño $k \times n$, donde k se puede elegir para que se cumpla $k << m$ y la reducción de la dimensión sea considerable.

Es posible ver la transformación PCA desde un punto de vista ligeramente diferente y considerar que $\mathbf{U}$ es una de las matrices de la descomposición SVD de la matriz de datos $\mathbf{X}$,

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \tag{5.3}$$

donde $\mathbf{U}$ y $\mathbf{V}$ son matrices cuyas columnas son ortonormales y su tamaño es $m \times r$ y $n \times r$ respectivamente. $\mathbf{\Lambda}$ es una matriz diagonal de tamaño $r \times r$ cuyos valores singulares son no negativos $\sigma_j, j = 1, \ldots, r$, y están ordenados de manera decreciente a lo largo de la diagonal, siendo r el rango de la matriz $\mathbf{X}$.

De la ecuación 5.3 se deriva que $\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{U}\mathbf{\Lambda}\mathbf{\Lambda}^T$ y $\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^T\mathbf{\Lambda}$, demostrando que las columnas de $\mathbf{U}$ son los vectores propios de $\mathbf{X}\mathbf{X}^T$ y las columnas de $\mathbf{V}$ son los vectores propios de $\mathbf{X}^T\mathbf{X}$.

También es posible la representación de la matriz $\mathbf{X}$ mediante la suma de r matrices de rango unidad,

$$\mathbf{X} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \tag{5.4}$$

Esto implica que los valores singulares nulos pueden ser ignorados puesto que no llevan consigo ningún tipo de información. La ecuación 5.5 también muestra que es posible aproximar $\mathbf{X}$ mediante los primeros k vectores propios de $\mathbf{U}$ y $\mathbf{V}$,

$$\mathbf{X} \approx \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{V}}^T \tag{5.5}$$

Un vector aleatorio y centrado $\mathbf{z}$ se dice que está blanqueado (*white*) si sus elementos están no correlados y tienen varianza unidad. Esto significa obviamente, que su matriz de covarianzas es igual a la matriz unidad $\mathbf{I}$. Como la operación de *whitening* puede realizarse mediante una decorrelación seguida de un escalado, puede utilizarse PCA con ese fin, y la ecuación 5.6 define la transformación de *whitening* sobre la matriz de datos original:

$$\widetilde{\mathbf{Z}} = \widetilde{\mathbf{\Lambda}}^{-1}\widetilde{\mathbf{U}}^T\mathbf{X} = \widetilde{\mathbf{V}}^T \tag{5.6}$$

5.1.1.2. ICA

ICA [79] intenta expresar los datos originales utilizando vectores aleatorios estadísticamente independientes. La matriz de datos aleatorios observada $\mathbf{X}$ se modela como

$$\mathbf{X} \approx \mathbf{AS} \tag{5.7}$$

donde $\mathbf{S}$ es la matriz que contiene los vectores aleatorios estadísticamente independientes y $\mathbf{A}$ es la matriz mezcla. El modelo anterior de la ecuación 5.7 también se puede expresar como:

$$\mathbf{S} = \mathbf{WX} \tag{5.8}$$

donde $\mathbf{W}$ es igual a la matriz pseudoinversa de $\mathbf{A}$, es decir, $\mathbf{W} = \mathbf{A}^{\dagger}$. En el modelo ICA, los vectores $\mathbf{w}_i$ se obtienen de forma que las filas de la matriz $\mathbf{S}$ tienen distribuciones no-gaussianas máximas y están mutuamente decorreladas. Una forma sencilla de conseguir ésto es primero, blanquear los datos según la ecuación 5.6, y después, buscar una proyección ortogonal no-normal ($\mathbf{R}$):

$$\mathbf{S} = \mathbf{R}^T \widetilde{\mathbf{\Lambda}}^{-1} \widetilde{\mathbf{U}}^T \mathbf{X} = \mathbf{R}^T \widetilde{\mathbf{Z}} \tag{5.9}$$

por lo tanto, en 5.9 se muestra que de hecho, ICA es una operación de blanqueo seguida de una rotación, y así, el modelo ICA puede expresarse también de la siguiente forma:

$$\mathbf{X} \approx \widetilde{\mathbf{U}} \widetilde{\mathbf{\Lambda}} \widetilde{\mathbf{V}}^T = \underbrace{\widetilde{\mathbf{U}} \widetilde{\mathbf{\Lambda}} \mathbf{R}}_{\mathbf{A}} \underbrace{\mathbf{R}^T \widetilde{\mathbf{V}}^T}_{\mathbf{S}} = \mathbf{AS} \tag{5.10}$$

Es necesario enfatizar el hecho que en el modelo ICA ninguna de las matrices de la transformación ($\mathbf{A}$ o $\mathbf{W}$) están restringidas a matrices ortogonales. La restricción (exacta [78] o aproximada [15], según la elección del algoritmo) es que la transformación sea decorrelante, esto es, que las filas de la matriz de datos independiente ($\mathbf{S}$) sean (exacta o aproximadamente) una transformación ortogonal de los datos blanqueados.

Además de medidas de no-gaussianidad, también es posible obtener el modelo ICA utilizando medidas de teoría de la información (e.g. negentropía), métodos tensoriales o por estimación de máxima verosimilitud (*ML estimation*). Entre los algoritmos más populares de ICA destacan *FastICA* [79, 78], *Infomax* [15] y su versión ampliada *Extended Infomax* [98]. Para el caso de *FastICA*, los datos son primero blanqueados y posteriormente rotados, tal como se vió en la ecuación 5.9. Sin embargo, *Infomax* no fuerza estrictamente una decorrelación de los datos completa. Por lo tanto, el resultado puede no ser exactamente una transformación ortonormal de los datos blanqueados, aunque se cumple para prácticamente todos los casos.

En las figuras 5.1 y 5.2 se ilustran gráficamente las transformaciones que producen los métodos de PCA, *whitening* e ICA (utilizando los algoritmos antes mencionados, *FastICA*

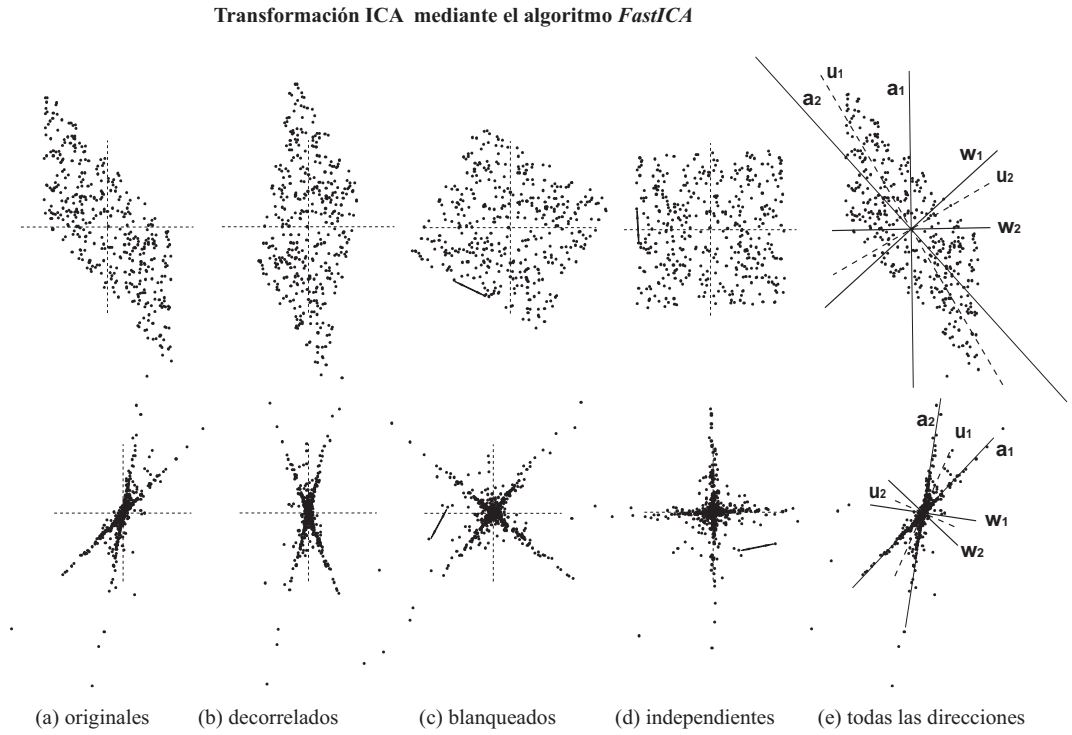


Figura 5.1: Dos ejemplos artificiales: un conjunto de datos subgaussiano (1ª fila) y un conjunto supergaussiano (2ª fila) (a), ambos transformados por PCA (b), *whitening* (c) y utilizando el modelo ICA a través del algoritmo *FastICA* (d). La última figura de cada fila (e) muestra el conjunto de datos original con las direcciones de PCA e ICA.

y *Extended Infomax*) sobre dos conjuntos de datos artificiales de sólo 2 dimensiones. En la primera fila de cada figura, los datos originales tiene una distribución uniforme sobre un paralelogramo (datos subgaussianos); mientras que los datos originales en la segunda fila siguen una distribución *sparse* (datos supergaussianos).

La figura 5.1 ha sido generada utilizando *FastICA*, por lo que la transformación ICA se muestra según la ecuación 5.9. En la fila superior, los datos originales tienen una distribución uniforme sobre un paralelogramo (figura 5.1(a)) pero las componentes no son independientes y es posible predecir el valor de una de ellas a partir del valor de la otra. El conjunto de datos decorrelados (transformados por PCA) se muestran en la figura 5.1(b), la dirección con la varianza máxima es el eje vertical, y el segundo eje principal es la dirección horizontal. Los datos blanqueados se muestran en la figura 5.1(c), la operación de *whitening* proporciona las componentes independientes salvo por una transformación ortogonal. Y, finalmente, el conjunto de datos independientes (transformados por ICA) aparece en la figura 5.1(d). El mismo proceso se muestra en la segunda fila para el caso de los datos con distribución supergaussiana. La figura 5.1(e) muestra los conjuntos de datos originales con las direcciones de ICA y PCA: se puede apreciar que la restricción

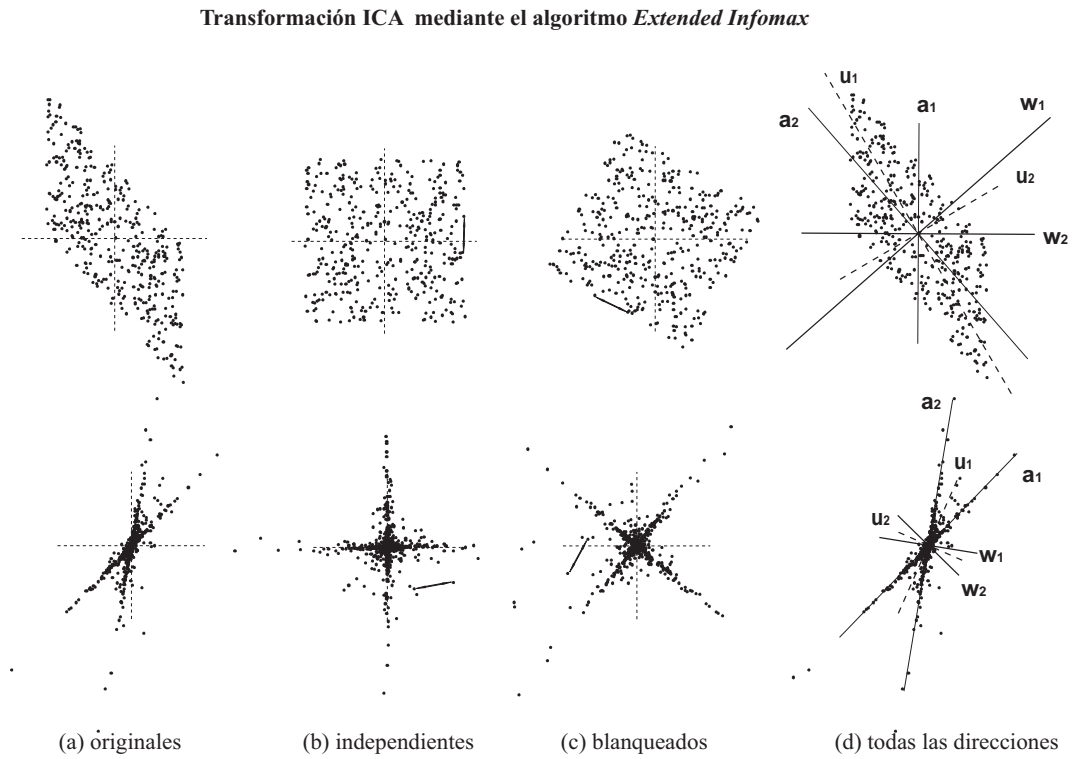


Figura 5.2: Dos ejemplos artificiales: un conjunto de datos subgaussiano (1ª fila) (a) y un conjunto supergaussiano (2ª fila), ambos transformados utilizando el modelo ICA a través del algoritmo *Extended Infomax* (b). Los datos blanqueados, se han copiado de la Fig.5.1 para su comparación. La última figura de cada fila (d) muestra el conjunto de datos original con las direcciones de PCA e ICA.

de ortogonalidad se cumple en la direcciones de PCA ($\mathbf{U}$), pero no en las direcciones de ICA ($\mathbf{W}$).

En los ejemplos anteriores puede apreciarse como la operación de blanqueo cambia las distancias entre puntos y por lo tanto, la distancia entre dos puntos no es la misma en los datos decorrelados que en los blanqueados. Sin embargo, la distancia entre dos puntos es igual en los datos blanqueados que en los independientes debido a que las componentes independientes son las blanqueadas seguidas de una rotación.

Con el fin de mostrar que las componentes independientes obtenidas por *Infomax* son también, aproximadamente, una transformación ortonormal de los datos blanqueados, la figura 5.2 muestra los mismos conjuntos de datos artificiales que la figura 5.1, pero en este caso, las componentes independientes han sido calculadas (Fig.5.2(b)) utilizando el algoritmo *Extended Infomax*[2] [98]. En el gráfico es posible observar que los datos in-

[2]Se ha utilizado el algoritmo de Infomax extendido[98] debido a que es capaz de manejar fuentes sub-

dependientes (Fig.5.2(b)) obtenidos con *Extended Infomax* son, aproximadamente, una versión rotada de los datos blanqueados (Fig.5.2(c), copiada de (Fig.5.1(c)) para su comparación).

Aunque la equivalencia anterior se cumple en los algoritmos de ICA más populares [15][78][79][98], no se mantiene para el algoritmo de Comon [37]. Esto es debido a que este algoritmo renormaliza las componentes de manera de las columnas de $\mathbf{A}$ tengan norma unidad. Esto produce que se modifiquen los valores de las varianzas de las componentes independientes (filas de $\mathbf{S}$). En este caso, la transformación ICA *no* es equivalente a una rotación de los datos blanqueados. Por lo tanto, las distancias y los ángulos entre los puntos independientes pueden cambiar respecto de la transformación de *whitening*. Sin embargo, exceptuando algunos trabajos de reconocimiento de objetos basados en la apariencia que utilizan el algoritmo de Comon [101][102][104], casi todas las comparativas entre PCA/ICA, como la mayoría de las citadas en la Tabla 2.3, utilizan *FastICA* o *Infomax* (o su versión extendida) para realizar ICA.

gaussianas y supergaussianas, puesto que el original [15] sólo era capaz de obtener las IC's de fuentes supergaussianas.

5.1.1.3. Diferentes arquitecturas para la descomposición ICA: ¿independencia en A o en S?

Trabajos previos [11][47] han descrito dos tipos distintos de *arquitecturas* para la descomposición ICA de un conjunto de imágenes de entrenamiento. La diferencia entre esas dos arquitecturas es una simple elección de dónde se busca la independencia. ¿Queremos que las columnas de $\mathbf{A}$ (imágenes base) sean independientes, o buscamos una descomposición donde los coeficientes (filas de $\mathbf{S}$) son mutuamente independientes? La primera corresponde a la arquitectura I, y la segunda a la arquitectura II (ver Fig. 5.3).

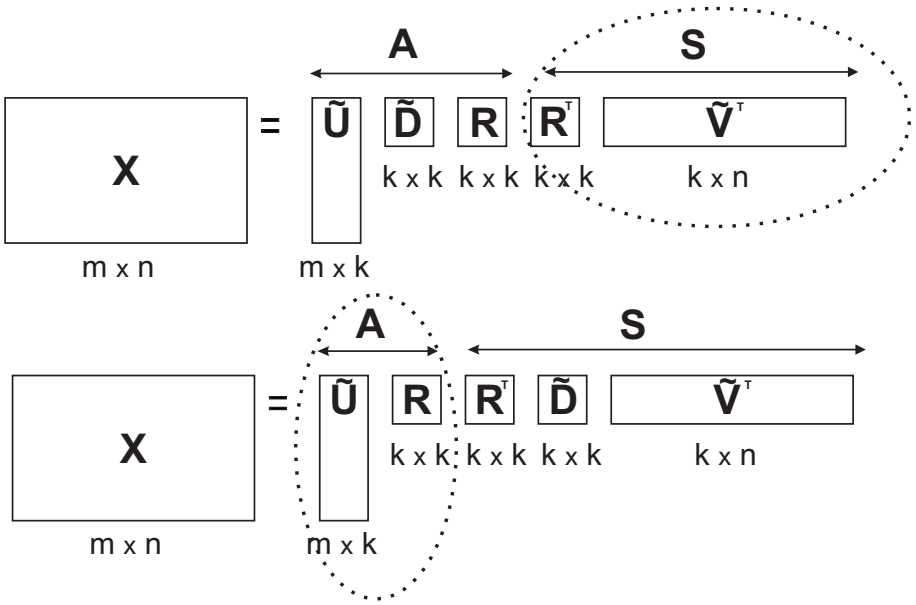


Figura 5.3: Diferentes arquitecturas para la descomposición ICA de un conjunto de imágenes: ¿queremos la independencia en $\mathbf{A}$ o en $\mathbf{S}$?

En ecuaciones, la cuestión es si escribimos $\mathbf{X} \approx \widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}\mathbf{R}\mathbf{R}^T\widetilde{\mathbf{V}}^T$, y optimizamos la matriz ortogonal $\mathbf{R}$ para hacer $\mathbf{S} = \mathbf{R}^T\widetilde{\mathbf{V}}^T$ tan independiente y no-gaussiano como sea posible, o si escribimos $\mathbf{X} \approx \widetilde{\mathbf{U}}\mathbf{R}\mathbf{R}^T\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{V}}^T$, y optimizamos $\mathbf{R}$ para que $\mathbf{A} = \widetilde{\mathbf{U}}\mathbf{R}$ sea tan independiente y no-gaussiano como sea posible.

Originalmente [11][47], las dos arquitecturas consistían simplemente en trabajar con la matriz de datos $\mathbf{X}$ o bien con su transpuesta $\mathbf{X}^T$ lo que equivale a utilizar la ecuación $\mathbf{X} \approx \widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}\mathbf{R}\mathbf{R}^T\widetilde{\mathbf{V}}^T$, y optimizar la matriz ortogonal $\mathbf{R}$ para que $\mathbf{S} = \mathbf{R}^T\widetilde{\mathbf{V}}^T$ sea independiente, o bien plantear $\mathbf{X}^T \approx \widetilde{\mathbf{V}}\widetilde{\mathbf{\Lambda}}\mathbf{R}\mathbf{R}^T\widetilde{\mathbf{U}}^T$, y optimizar $\mathbf{R}$ para que $\mathbf{S} = \mathbf{R}^T\widetilde{\mathbf{U}}^T$ sea independiente. La descripción anterior es equivalente a la primera, pero intercambiando los papeles de $\mathbf{A}$ y $\mathbf{S}$ en el modelo ICA de la arquitectura I. Este otro enfoque se ilustra gráficamente en la figura 5.4.

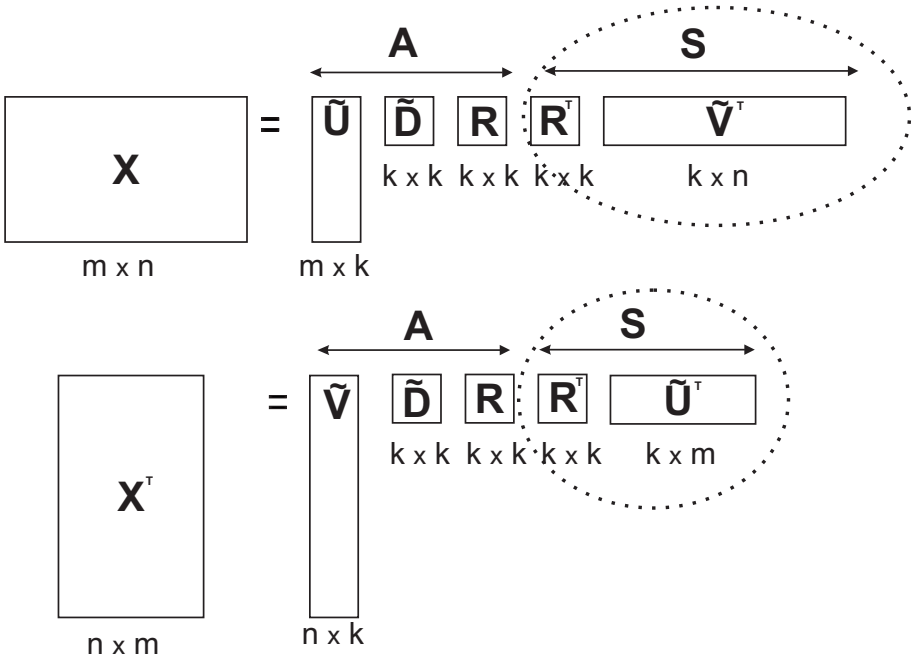


Figura 5.4: Diferentes arquitecturas para la descomposición ICA de un conjunto de imágenes: ¿queremos la independencia en $\mathbf{A}$ o en $\mathbf{S}$?

En cualquier caso, las componentes independientes, que es la matriz que ha sido optimizada para la independencia, es siempre una transformación ortogonal de los datos blanqueados proporcionados por PCA. Como se discutirá en la sección siguiente, esto implica que si se utiliza un clasificador invariante rotacional, no hay ninguna ventaja en realizar la optimización de $\mathbf{R}$.

5.1.1.4. Clasificadores invariantes ante rotación

La mayoría de los clasificadores más utilizados son invariantes a la rotación del espacio donde residen los datos. Intuitivamente, parece lógico utilizar clasificadores invariantes rotacionales si no se tiene un conocimiento previo de la estructura del espacio. Ejemplos típicos de este tipo de clasificadores son todos aquellos basados en distancias euclídeas, o basados en ángulos, entre los puntos del conjunto de datos. Como se mostró en la ecuación 5.9, y se discutió en la sección precedente, las componentes independientes son el resultado de una rotación sobre los datos blanqueados por SVD. Por lo tanto, un clasificador invariante ante rotación funcionará de igual forma si su entrada es el conjunto de las componentes independientes como si es el conjunto de los datos blanqueados. En conclusión, bajo estas circunstancias ICA no proporciona ninguna ventaja sobre PCA.

Sin embargo, existen un par de escenarios en los que ICA puede proporcionar un resultado diferente (y por tanto posiblemente mejor) que PCA:

- Si el clasificador utilizado no es invariante ante rotación, pueden existir ventajas al realizar la rotación que ICA proporciona. Por ejemplo, en [26] se demuestra que utilizando datos independientes un clasificador bayesiano tipo *Naive Bayes* proporcionaría mejores resultados que si los datos no se preprocesaran mediante ICA.

- Si posteriormente a la transformación de ICA se utiliza un método de selección de características para seleccionar un subconjunto de todas las componentes independientes, entonces los subespacios ICA y PCA escalado (*whitening*) pueden diferir, y por tanto los resultados de la clasificación también serán distintos para cada uno de los dos métodos.

En conclusión, si no se realiza una selección de características y se utiliza un clasificador invariante ante rotación, es absolutamente innecesario realizar la costosa transformación de ICA en lugar de utilizar simplemente la transformación clásica de PCA. La figura 5.24 muestra una comparativa de los tiempos de cómputo de los subespacios de los métodos considerados en esta tesis: PCA, ICA, NMF y RP. En ella podemos apreciar el alto coste computacional que lleva asociado ICA en comparación con PCA, aún cuando el algoritmo utilizado para obtener las componentes independientes es *FastICA*, el de menor coste computacional que existe actualmente [79].

5.1.1.5. Selección de subconjuntos de componentes

En las secciones previas, se ha mostrado que, aplicado típicamente, no existe ningún motivo para preferir utilizar las componentes de ICA sobre los datos blanqueados. Sin embargo, el rendimiento de ICA y PCA pueden diferir cuando se utiliza para la clasificación un subconjunto de las componentes. Para ilustrar este hecho, se ha utilizado un conjunto de datos artificial similar al conjunto subgaussiano de la figura 5.1, pero asignando dos clases diferentes al conjunto de los puntos en dos casos extremos. El ejemplo se muestra en la figura 5.5. En ambos gráficos, están trazadas las direcciones de los vectores propios, $\mathbf{U}$ y las direcciones de ICA, $\mathbf{W}$ y $\mathbf{A}$, del conjunto de datos. Como ambas técnicas son métodos no supervisados, las direcciones que se obtienen son independientes de las clases existentes en el conjunto de los datos, y por lo tanto, son iguales en los dos ejemplos extremos. En la figura 5.5(a) es fácil apreciar como las clases están perfectamente separadas utilizando solo la proyección de los puntos originales en la dirección ICA $\mathbf{w}_1$, mientras que en la figura 5.5(b), las clases están mejor separadas utilizando la dirección del vector propio $\mathbf{u}_2$. Por tanto, en cada uno de los dos ejemplos extremos, una selección de características puede ayudar a reducir la dimensión y a mejorar los resultados de la clasificación. Si no se realiza una selección de características previa a la etapa de clasificación, ICA y los datos transformados por PCA blanqueados producen los mismos resultados de clasificación si el clasificador utilizado es invariante ante rotación, como vimos en el apartado anterior.

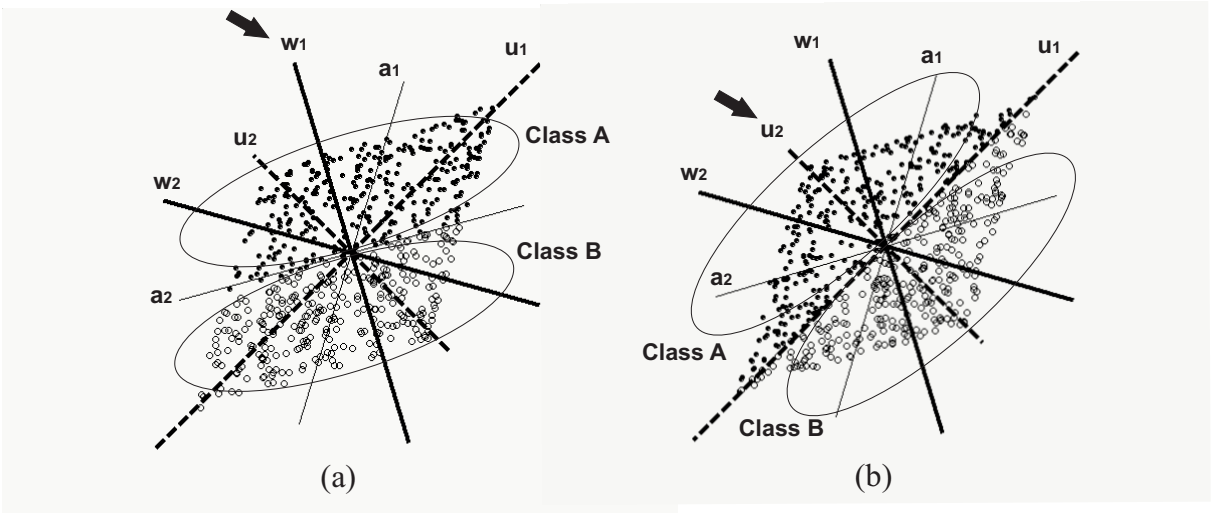


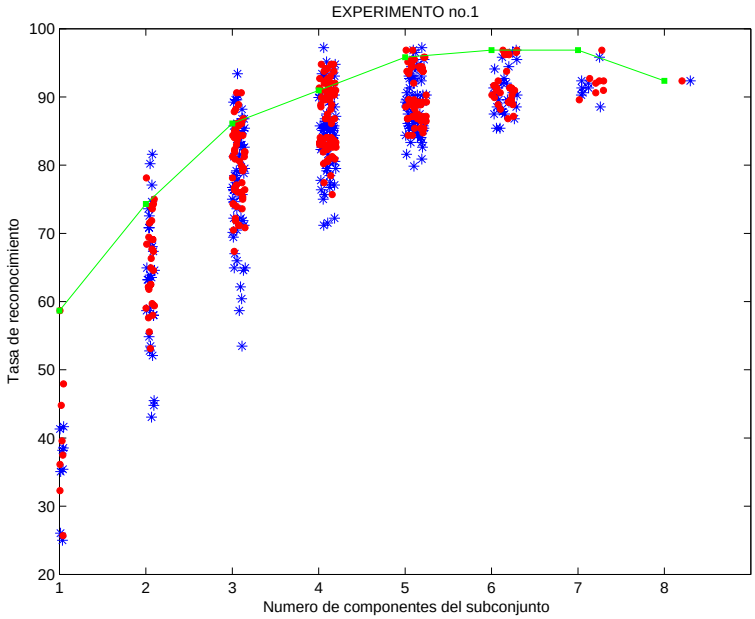
Figura 5.5: La *mejor* selección de características puede ser diferente según la distribución de las clases en el conjunto de datos. En la figura 5.5(a) las clases están perfectamente separadas utilizando sólo la proyección sobre la dirección ICA definida por el vector $\mathbf{w}_1$, mientras que en la figura 5.5(b), las clases están mejor separadas utilizando la dirección del eigenvector $\mathbf{u}_2$

Para confirmar la idea anterior: que si se utiliza un proceso previo de selección de características, ICA puede producir tasas de éxito en el reconocimiento mayores que las de PCA y viceversa, se ha realizado un sencillo experimento con los objetos de la COIL-100 haciendo uso de un proceso de selección de características *exhaustivo* [48]. Este tipo de selección de características denominado *búsqueda exhaustiva* o *búsqueda por fuerza bruta* es un algoritmo óptimo, es decir, garantiza la optimalidad del proceso de selección, puesto que examina todas las combinaciones posibles de P variables extraidas de las M iniciales. El número de posibles combinaciones es:

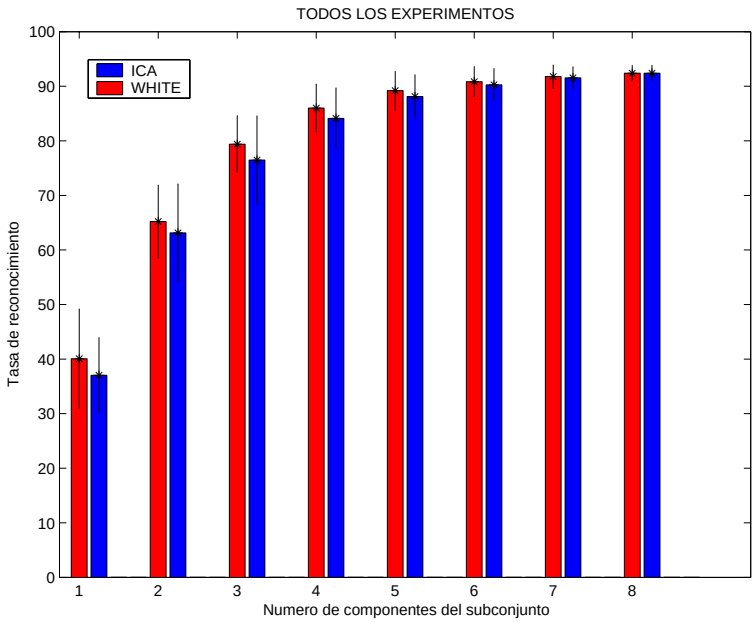
$$\binom{M}{P} = \frac{M!}{(M-P)!P!} \tag{5.11}$$

y resulta evidente que incluso para valores moderados de M y P esta búsqueda resulta costosísima.

Se utilizaron los 8 primeros objetos de la COIL-100 para comparar el funcionamiento de las transformaciones de *ICA* y *whitening* utilizando las imágenes en niveles de gris y una arquitectura de subespacio universal, es decir, una imagen de cada objeto es utilizada para la generación del subespacio ICA o PCA, y el resto son utilizadas para el test del clasificador elegido. El número de objetos del sistema de reconocimiento es tan reducido, sólo 8, debido a que se ha utilizado el proceso de selección de características exhaustivo, siendo como hemos visto el de mayor coste computacional puesto que genera todos los posibles subconjuntos de características. El clasificador utilizado fue el vecino más cercano tipo 1-NN con distancia euclídea, siendo por lo tanto, invariante ante rotaciones del espacio de datos utilizado. Los datos originales pueden ser representados sin pérdida de información utilizando subespacios PCA e ICA de tan solo 8 componentes, y en el caso que se utilicen todas ellas el resultado de ambos métodos debe ser el mismo; pero si se seleccionan subconjuntos de componentes, los resultados de reconocimiento pueden variar en función de la naturaleza de los datos. De esta manera, se han realizado los experimentos de entrenamiento y test asociados a todos los posibles subconjuntos de componentes (selección de características exhaustiva).



(a) Resultados de un solo experimento



(b) Estadística de 10 experimentos diferentes (*cross-validaton*)

Figura 5.6: Resultados de clasificación para los datos de ICA y PCA blanqueados utilizando subconjuntos de características.

En la figura 5.6(a) aparecen los resultados de uno de los experimentos utilizando selección exhaustiva. En el gráfico se han trazado en rojo las tasas de éxito en el reconocimiento de los subconjuntos de características seleccionados de los datos transformados por *whitening* y en azul los transformados por ICA. Al utilizar 8 objetos con la arquitectura de subespacio universal con una sola imagen de entrenamiento, existirán como máximo 8 vectores que definan la transformación, por lo que para el proceso de selección exhaustiva $M = 8$, y, P irá variando desde 1 a 8. La tabla 5.1 muestra el número de subconjuntos a testear con componentes distintas según el numero de componentes.

♯ componentes	♯ subconjuntos
1	8
2	28
3	56
4	70
5	56
6	28
7	8
8	1

Tabla 5.1: Número de subconjuntos a testear en la selección exhaustiva de características.

De esta forma, si la dimensión elegida es 1, existirán 8 posibles casos, si el número de componentes es 2, serán 28 los subconjuntos de características a testear, etc. En la figura 5.6(a) se han dibujado el resultado de cada uno de estos casos introduciendo un ligero ruido aleatorio en el eje horizontal (*jitter*) de cada muestra para que no se solapen los puntos y se aprecien mejor los resultados.

Además, en el mismo gráfico, se muestran en color verde las tasas de reconocimiento de los datos transformados por *whitening* utilizando subconjuntos formados por las componentes ordenadas de mayor a menor varianza, es decir, utilizando una sola componente se elige la asociada al valor propio de mayor valor, utilizando dos componentes, las asociadas a los dos valores propios de mayor valor y así hasta el caso de 8 componentes.

En el gráfico de la figura 5.6(a) se puede apreciar como algunos subconjuntos presentan mejores tasas de éxito en función del método de extracción utilizado. Por ejemplo, para el caso de una sola componente se aprecia como las direcciones de PCA separan mejor las clases que las de ICA, siendo la mejor de todas la dirección de máxima varianza. Sin embargo, si se utilizan subconjuntos de 2, 3 o 4 componentes, ICA proporciona tasas de éxito mayores que las de PCA; pero para los casos de 5, 6 y 7, los resultados de ambas técnicas son muy similares, y en el caso de 8 (el total de componentes) el resultado es exactamente el mismo, tal como se había previsto.

La estadística de realizar una prueba de *cross-validation* para el experimento de la figura 5.6(a) se muestra en la figura 5.6(b). La prueba de *cross-validation* consiste en la repetición del mismo experimento 10 veces, variando los patrones de entrenamiento y test, y generando cada vez los subespacios PCA e ICA. Los resultados indican que para la mayoría de los casos de subconjuntos de P componentes, la técnica de PCA más blanqueo produce una tasa de éxito en el reconocimiento *media* superior a la de ICA, pero sin embargo, la desviación típica de ICA engloba los mejores resultados de PCA (*whitening*).

En conclusión, este sencillo experimento nos muestra la misma idea que el gráfico de las dos clases de datos artificiales en sus casos extremos: si se realiza una selección de características, PCA(*whitening*) e ICA pueden diferir en los resultados de clasificación pero el que una técnica sea superior o no a la otra depende sólamente de la naturaleza de los datos.

5.1.2. Posibles fuentes de error en la comparación de PCA e ICA

En los apartados anteriores se ha demostrado analítica y experimentalmente que si se utilizan el mismo número de componentes en los métodos de ICA y PCA, sin realizar una selección de características previa a la etapa del clasificador, y si éste es invariante rotacional, los resultados de ambos métodos deben coincidir.

Sin embargo, en la literatura existen diversos trabajos en los que las tasas de éxito en el reconocimiento de ICA superan a las de PCA aún sin utilizar métodos de selección de características. Estos resultados pueden deberse a posibles fuentes de error en la aplicación de PCA e ICA que pasamos a enumerar y describir[3]:

1. Los algoritmos de cómputo de las componentes independientes más extendidos son *FastICA* e *Infomax*. Ambos son métodos numéricos, y aún siendo teóricamente equivalentes, poseen propiedades de convergencia distintas, lo que da lugar a que las componentes independientes que se obtienen no son siempre equivalentes, siendo por tanto una fuente de error en la posible comparativa entre ICA y PCA. A esto cabe añadir que para el caso de FastICA, los datos son primero blanqueados y posteriormente rotados, tal como se vió en la ecuación 5.9. Sin embargo, Infomax no fuerza estrictamente una decorrelación de los datos completa. Por lo que el resultado puede no ser exactamente una transformación ortonormal de los datos blanqueados; aunque, sin embargo, esto se mantiene para prácticamente todos los casos en los que la presencia de puntos *outliers* es despreciable. Además, los primeros trabajos de reconocimiento basado en la apariencia se realizaron con el algoritmo *Infomax* (1995), puesto que *FastICA* es de aparición posterior (1997) y la mayoría de los artículos que realizan las comparativas utilizan el algoritmo de Bell y Sejnowski [11] [44] [47] [149].

2. Si se utiliza el modelo ICA como método de extracción de características para un sistema de reconocimiento de objetos basado en la apariencia, las componentes que se obtendrán, en función del algoritmo que se utilice para su estimación, serán, como hemos visto, y con mayor o menor precisión, según la convergencia del algoritmo, una rotación de los datos blanqueados. La operación de *whitening* se define en la ecuación 5.6 y en la figura 5.1(c) se muestra el resultado de blanquear un conjunto de datos. Si se realiza una comparativa entre PCA e ICA y no se utilizan los datos blanqueados, sino los obtenidos a través de la ecuación 5.1 sin el escalado (figura 5.1(b)), es evidente que los resultados de ambos métodos difererirán puesto que la distancia entre dos puntos cualesquiera es diferente para cada subespacio. Luego el utilizar la transformación definida en la ecuación 5.1 en vez de la 5.6 puede deteriorar la comparativa con ICA.

3. Otra posible fuente de error en las comparativas entre ICA y PCA es el hecho de centrar o no los datos antes de realizar la proyección en el subespacio elegido. Con

[3] Algunos de los resultados que aquí se presentan también se pueden consultar en [160] y [161].

el método de PCA, queda claro que para no perder información si se trabaja con los datos proyectados en el subespacio PCA es necesario centrar los datos antes de realizar el cómputo del eigenespacio y la proyección posterior; pero el modelo ICA no necesita de la restricción de trabajar con datos centrados, luego no queda claro si al utilizar la transformación ICA como método de extracción de características se han de centrar los datos o no. Está claro que si se quiere realizar una comparativa estricta entre ambos métodos los datos de entrada para ambos han de ser iguales, centrados o no, en otro caso, la comparativa no es del todo adecuada.

5.1.2.1. PCA vs. whitening

Como se ha descrito anteriormente, si se realiza una comparativa entre PCA e ICA y no se utilizan los datos blanqueados, sino los obtenidos a través de la ecuación 5.1 sin el escalado (figura 5.1(b)), es evidente que los resultados de ambos métodos difererirán puesto que la distancia entre dos puntos cualesquiera es diferente para cada subespacio. Luego el utilizar la transformación definida en la ecuación 5.1 en vez de la 5.6 puede deteriorar la comparativa con ICA, puesto que lo que en realidad se está haciendo es comparar los resultados de PCA (Eq. 5.1) con los de whitening (eq. 5.6). En este apartado se muestran los resultados de realizar esta comparativa entre PCA *vs. whitening.*

Comparativa con subespacios universales

Para obtener una comparativa del funcionamiento de PCA y *whitening* con subespacios universales se calcularon diversos subespacios utilizando la vista principal de cada uno de los objetos con conjuntos de objetos diferentes (desde 10 a 100 objetos de la COIL-100) y utilizando diversos canales de características (niveles de gris, bordes de la imagen e histogramas de color). El resto de las vistas se utilizaron como ejemplos de test y el clasificador utilizado fue el vecino más cercano (k-NN con k=1 y la norma L2 como medida de distancia).

En las figuras 5.7 y 5.8 se muestra un resumen de los resultados experimentales de ambos métodos de extracción de características con subespacios universales utilizando el canal de apariencia I (niveles de gris): se aprecia que ambas técnicas funcionan de manera muy similar. En la figura 5.7 aparece la tasa de éxito en el reconocimiento utilizando PCA y *whitening* frente al número de objetos presentes en el sistema de reconocimiento y utilizando sólo 10 componentes para representar cada subespacio. Según aumenta el número de objetos, se reduce el porcentaje de éxito en la clasificación para ambas técnicas. En la figura 5.8 se muestra la influencia del número de componentes (dimensión) del subespacio en el reconocimiento correcto de los objetos utilizando un conjunto de 20 objetos y variando la dimensión del subespacio de 2 a 20 componentes. Tal como se esperaba, el porcentaje de aciertos en el reconocimiento se incrementa conforme aumenta la dimensión del subespacio universal. Experimentos con conjuntos de objetos mayores (hasta 100 objetos) y empleando el resto de canales muestran resultados similares.

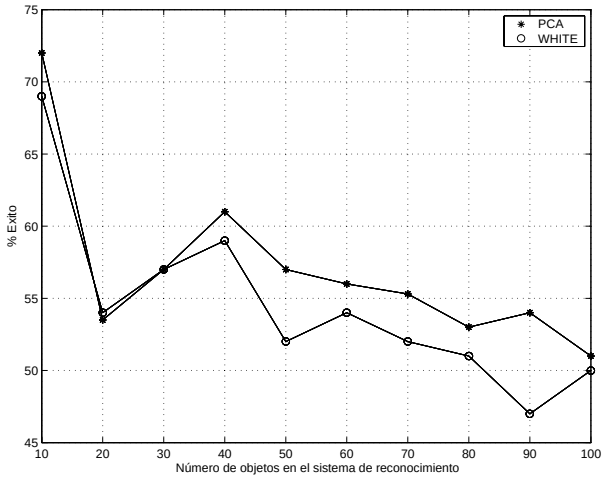


Figura 5.7: Subespacios universales: tasa de éxito en el reconocimiento para PCA y *whitening* utilizando solo 10 componentes.

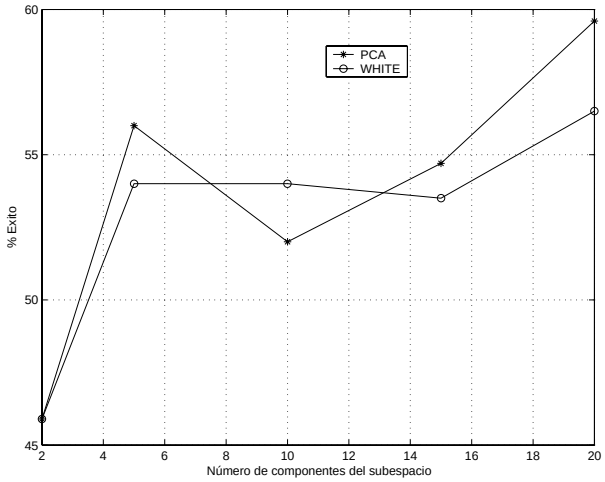
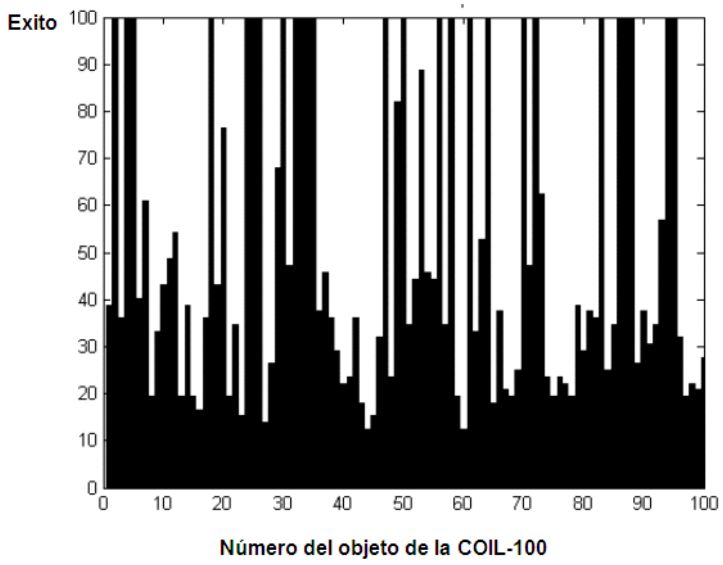


Figura 5.8: Subespacios universales: tasa de éxito en el reconocimiento para PCA y *whitening* en relación con el número de componentes del subespacio.

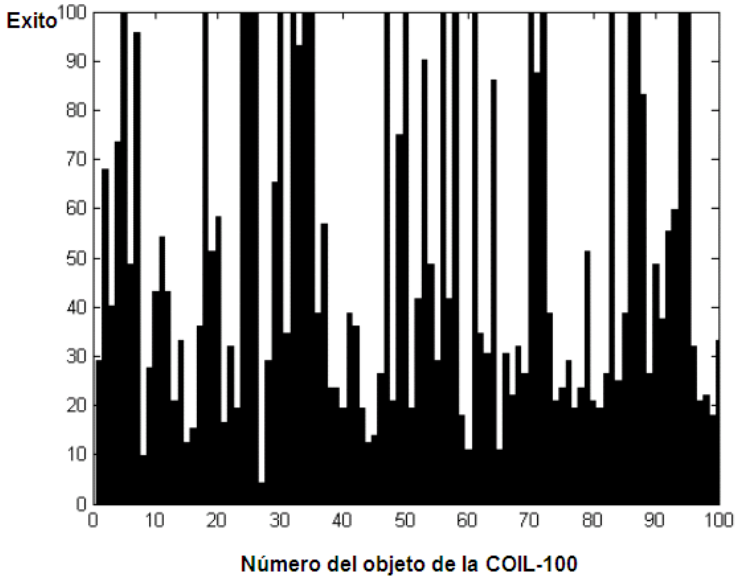
Como ya vimos en los experimentos del capítulo 4, los subespacios universales funcionan de manera excepcional con objetos simétricos, es decir, aquellos cuyas vistas o poses son muy parecidas entre sí; pero no funcionan tan bien con objetos no muy simétricos. Este fenómeno se muestra en las figuras 5.9(a) y 5.9(b), en donde se presentan las tasas de éxito en el reconocimiento de cada uno de los 100 objetos de la COIL-100: los objetos simétricos tienen un tasa de éxito en el reconocimiento del $100\,\%$ en ambas técnicas de extracción de características, mientras que los objetos no-simétricos se confunden totalmente. En estas gráficas también es posible apreciar que los datos blanqueados proporcionan tasas de éxto ligeramente peores que las de PCA, pues hay 26 objetos simétricos $(100\,\%)$ con PCA y sólo 21 con *whitening*. Este mismo hecho se ilustra también en las figuras 5.10(b) y 5.10(c), pero sólo para cuatro objetos simétricos (2, 4, 5 y 18) y otros cuatro no-simétricos (1, 3, 6 y 14), mostrando el error[4] en el eje Y en función del tamaño total del sistema (número de objetos a reconocer). Es apreciable como el error en el reconocimiento para los objetos simétricos es casi nulo e independiente del tamaño del sistema, mientras que para los objetos no-simétricos, el error aumenta conforme aumenta el número total de objetos en el sistema, pues existe mayor posibilidad de confusión entre los objetos.

Por lo tanto, bajo este contexto, los 100 objetos de la librería COIL-100 utilizando una arquitectura de subespacio universal con sólo una imagen por objeto para la generación del subespacio, los resultados de PCA son ligeramente superiores a los del proceso de *whitening*. Lo que puede justificar los resultados presentados por algunos trabajos previos en los que se sugería que PCA era superior a ICA.

[4]El error en el reconocimiento es igual al $100\,\% -$ tasa de éxito



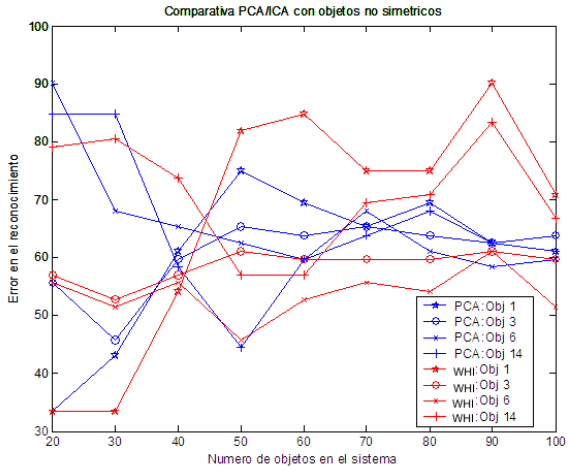
(a)



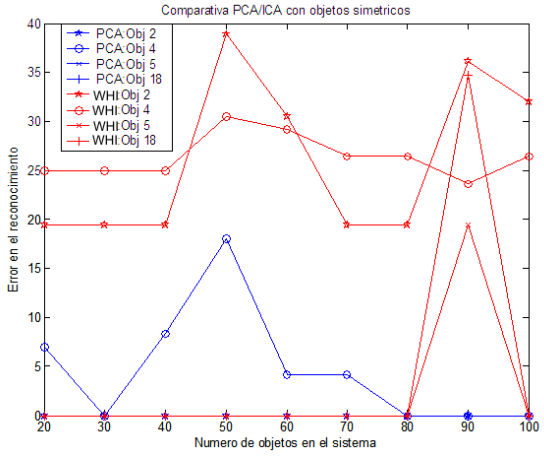
(b)

Figura 5.9: (a) Objetos simétricos con PCA: {2, 4, 5, 18, 24, 25, 26, 30, 32, 33, 34, 35, 47, 50, 56, 58, 61, 64, 70, 72, 83, 86, 87, 88, 94, 95}; (b) Objetos simétricos con *whitening*: {5, 18, 24, 25, 26, 30, 32, 34, 35, 47, 50, 56, 58, 61, 70, 72, 83, 86, 87, 94, 95}

(a)



(b)



(c)

Figura 5.10: (a) 4 objetos *simétricos* y 4 *no-simétricos* de la COIL-100;(b) error para PCA y *whitening* para los 4 objetos *no-simétricos*; (c) error para PCA y *whitening* para los 4 objetos *simétricos*.

Para confirmar los resultados anteriores de la comparativa de la separación entre clases, en cada uno de los dos tipos de subespacios, se ha definido un índice de separabilidad trivial que representa el grado de separación de una cierta clase y puede expresarse de la siguiente forma:

$$\alpha_t = \log\left[\frac{\text{máx}(d_t)}{\text{mín}(D_t)}\right] \tag{5.12}$$

donde α_t es el índice de separabilidad trivial para la clase t, d_t es la norma L2 entre dos proyecciones pertenecientes a la misma clase t (distancias entre elementos de la misma clase) y D_t es la norma L2 entre dos proyecciones donde sólo una de ellas pertenece a la clase t (distancias entre elementos de clases distintas). Valores altos de α_t significan que las clases son fácilmente separables y en particular, si es mayor que 0, las clases pueden separarse trivialmente con cualquier clasificador lineal sencillo. La figura 5.11 muestra una representación bidimensional del índice de separabilidad trivial.

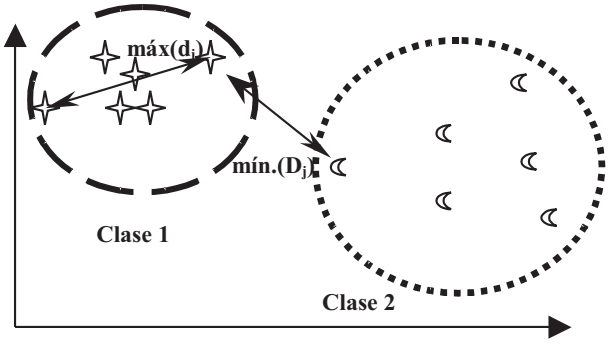
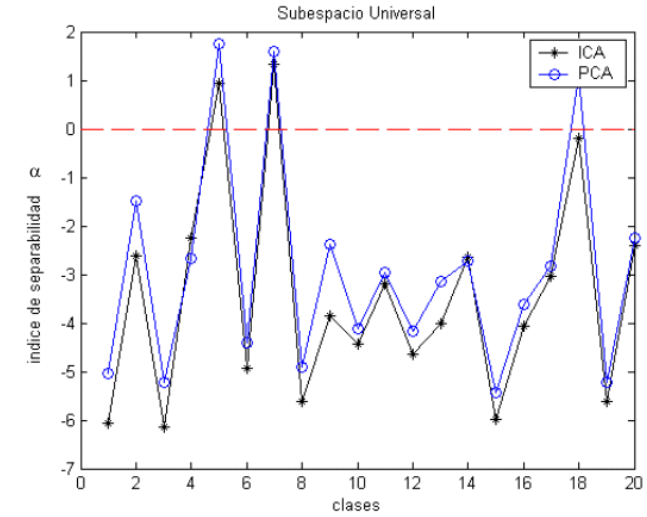


Figura 5.11: Representación bidimensional del índice de separabilidad trivial.

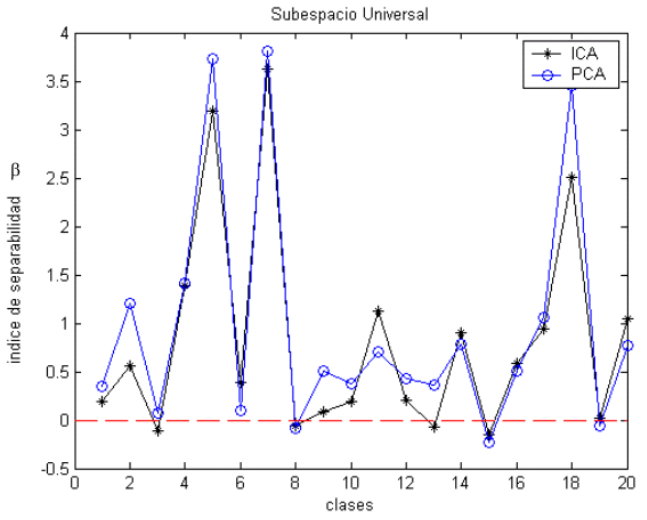
También, a partir de la ecuación 5.12 se puede utilizar otro índice de separabilidad menos restrictivo utilizando las medias de las clases:

$$\beta_t = \log\left[\frac{media(d_t)}{media(D_t)}\right] \tag{5.13}$$

Los resultados de separabilidad entre clases (también sólo para el canal I) se resumen en las figuras 5.12(a) y 5.12(b): PCA y *whitening* producen clases con separabilidad muy semejantes, aunque el primer método parece presentar ligeramente una separación mayor. En la figura 5.12(a) se utiliza α_j para estudiar la separación entre clases, en este caso sólo los objetos 5, 7 y 18 están separados del resto, de forma que cualquier clasificador simple los identificaría sin gran dificultad. Si utilizamos las medias de las clases, β_j, la separación intraclases mejora, como se muestra en la figura 5.12(b).



(a)



(b)

Figura 5.12: Indices de separabilidad, α_t (a), y β_t (b) para PCA y *whitening* con los 20 primeros objetos de la COIL-100 y arquitectura universal.

Comparativa con subespacios objeto modificados

Murase en [118] utiliza los subespacios objeto generados con PCA para averiguar la pose o vista del objeto y no para su identificación o reconocimiento, sin embargo, los subespacios objeto también se pueden utilizar para la identificación del objeto en sí como hemos propuesto en esta tesis en el apartado 4.2 . Además, en este caso, si se emplea el blanqueo como método de extracción de características, en lugar de sólo PCA, la técnica del subespacio objeto modificado aumenta considerablemente su tasa de éxito en el reconocimiento.

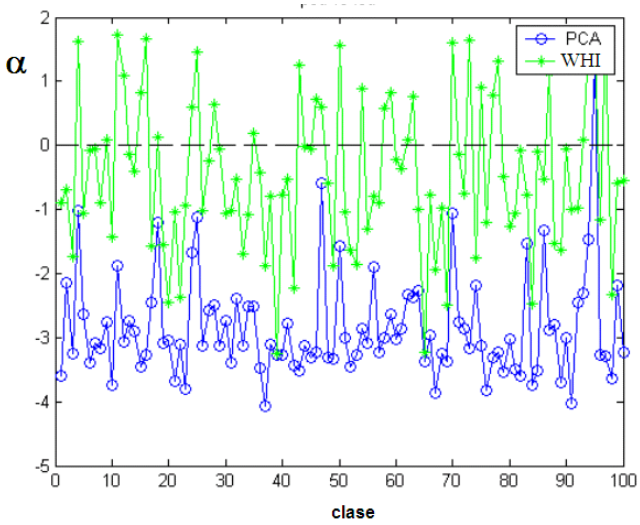
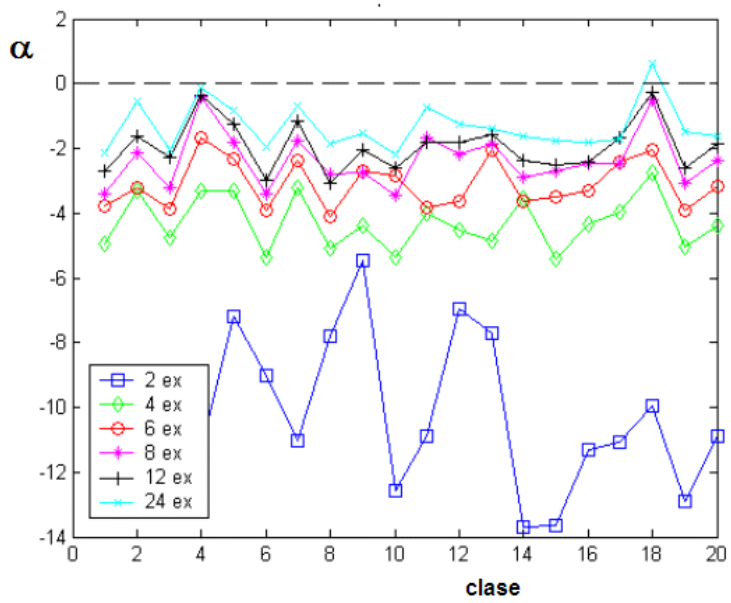


Figura 5.13: Índice de separabilidad para los subespacios objeto modificados PCA y *whitening* para la librería de objetos completa en el canal I. Los subespacios objeto se han obtenido utilizando solo 8 imágenes de entrenamiento por objeto.

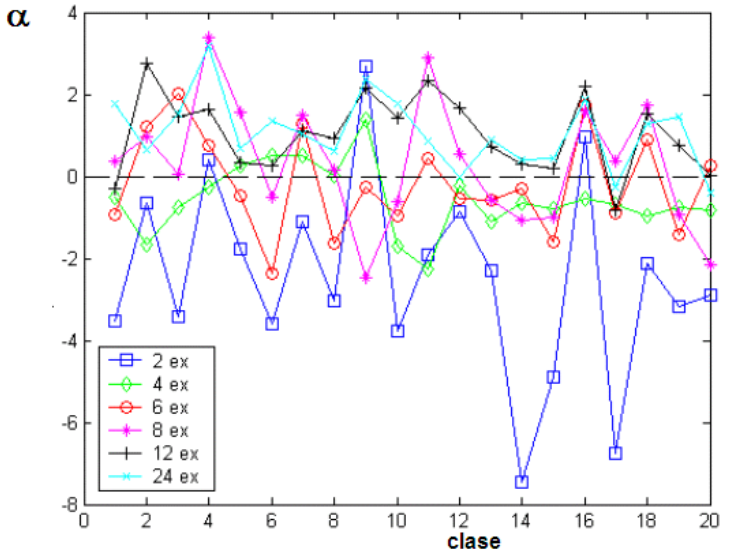
La figura 5.13 muestra el índice de separabilidad trivial para todos los objetos (clases) de la COIL-100 utilizando subespacios objeto modificado: la mayoría de las clases obtenidas mediante blanqueo están muy cerca o incluso sobrepasan el valor de separación trivial de $\alpha_t = 0$. Sin embargo, en el caso de PCA, la mayoría de los objetos están por debajo de este valor, lo que significa que no es tan sencillo su separación. Este hecho se muestra de nuevo en los gráficos de las figuras 5.14(a) y 5.14(b), que muestran los valores del índice de separabilidad trivial en función del número de imágenes utilizadas en la generación del subespacio (desde 2 a 24). Puede apreciarse como el indice de separabilidad es negativo para PCA con 24 imágenes de entrenamiento mientras que para los datos blanqueados es siempre mayor y algunos clases son trivialmente separables con solo 4 componentes.

Por lo tanto, para este canal de características (I) con los primeros 20 objetos de la

COIL-100, parece que sería mejor utilizar el proceso de blanqueo. Sin embargo, con datos de otra naturaleza este hecho deja de cumplirse. Las figuras 5.15(a) y 5.15(b) muestran el mismo experimento pero utilizando un canal de color, específicamente, el histograma RGB de la imagen. Para este canal, los objetos no están tan claramente separados como en el caso del canal I para el método de *whitening*.

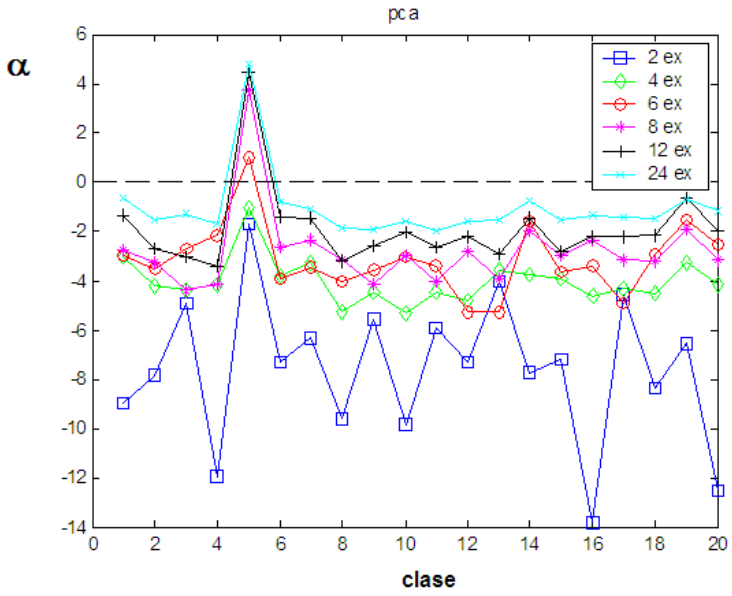


(a)

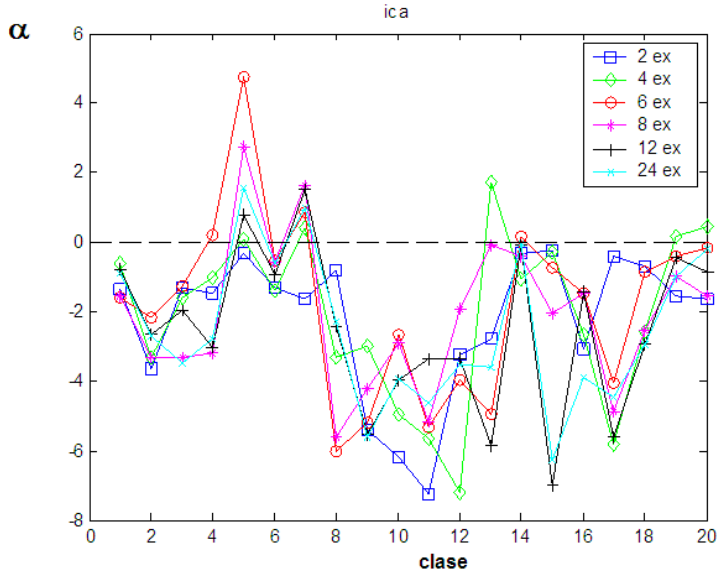


(b)

Figura 5.14: Índice de separabilidad para los subespacios objeto modificados de los 20 primeros objetos de la COIL100 (canal I) utilizando subespacios de diferentes dimensiones (desde 2 a 24 componentes): (a) PCA, (b) *whitening*.



(a)



(b)

Figura 5.15: Índice de separabilidad para los subespacios objeto modificados de los 20 primeros objetos de la COIL100 (canal histograma RGB) utilizando subespacios de diferentes dimensiones (desde 2 a 24 componentes): (a) PCA, (b) *whitening*.

5.1.2.2. Influencia de la normalización

Con el objetivo de verificar el efecto que produce el mantener o no la media de los datos antes de realizar la transformación de PCA o ICA, se han realizado una serie de experimentos de reconocimiento con un conjunto de objetos de la COIL-100 que persiguen confirmar que la operación de centrado afecta a los resultados, y por tanto, puede influenciar una comparativa entre PCA/ICA.

Se han realizado pruebas tanto utilizando la arquitectura universal como la objeto. Primero, se describen los resultados obtenidos con la arquitectura universal.

Para estudiar la influencia de la normalización de los datos en los resultados de reconocimiento, se han construido cuatro sistemas de reconocimiento PCA con arquitectura universal utilizando diversas normalizaciones de los datos de entrada:

- sin normalizar
- normalización global o centrado de datos
- normalización local
- normalización global y local

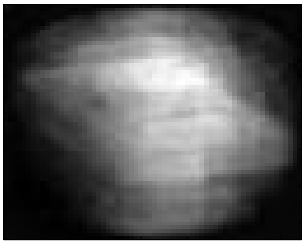
La figura 5.16(a) muestra las imágenes de entrenamiento en niveles de gris de los objetos utilizados para la obtención del subespacio universal en los distintos casos de normalización, y en la figura 5.16(b) aparece la imagen media, μ, del conjunto de las N imágenes de entrenamiento $\mathbf{X}$:

$$\mu = \frac{1}{N}\sum_{j=1}^{N}\mathbf{X} = \frac{1}{N}\sum_{j=1}^{N}x_{ij} \tag{5.14}$$

La normalización global o centrado de datos, consiste en restar a cada imagen la imagen media μ. De este modo se elimina la información común a todas las imágenes de todos los objetos que, por se común, no resulta útil para el reconocimiento.

La normalización local consiste en desplazar los niveles de gris de la imagen de modo que queden centrados en cero, o dicho de otro modo, en restar a cada píxel de la imagen la media de todos los píxeles (media local).

(a) Imágenes de entrenamiento de los objetos de la COIL-100 utilizados para el experimento de normalización.



(b) Imagen media, μ.

Figura 5.16: Experimento de normalización

Se han realizado experimentos de comparación de capacidad de reconocimiento para distintos niveles de compresión de información. En todos los casos se eligen las componentes principales de mayor varianza, y el rango de compresiones varía desde el caso de máxima compresión, donde sólo se utilizan 6 componentes hasta el caso de mínima compresión, donde se utilizan 20 componentes. Para cada nivel de compresión se comparan las tasas de error en reconocimiento utilizando como clasificador el método del vecino más cercano (k-NN), con k=1.

La figura 5.17 muestra un resumen de todos los resultados obtenidos en función del nivel de compresión de la información. Obviamente, las tasas de error disminuyen al reducirse la compresión, dado que se utiliza más información para el reconocimiento. En cuanto al comportamiento en función de las normalizaciones efectuadas, las diferencias son menores de lo esperable. Aparentemente, los mejores resultados se obtienen sin normalización, particularmente cuando el número de componentes principales utilizado es elevado. No obstante, es posible apreciar cómo las diferencias entre los distintos métodos no son significativas en general, de acuerdo con los valores de las desviaciones típicas mostrados en el gráfico. Las tasas de error se han obtenido trabajando sobre un subconjunto de 20 objetos pertenecientes a la librería COIL-100, cada uno de ellos representado mediante 72 imágenes (1 para entrenamiento y el resto para test). Se ha realizado un experimento de validación cruzada de 10 divisiones, repetido 10 veces para una mayor fiabilidad de los resultados.

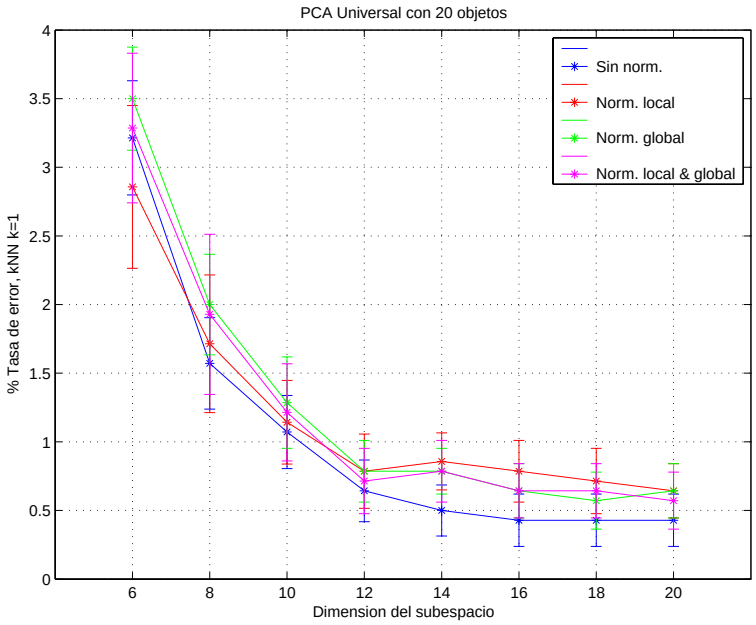
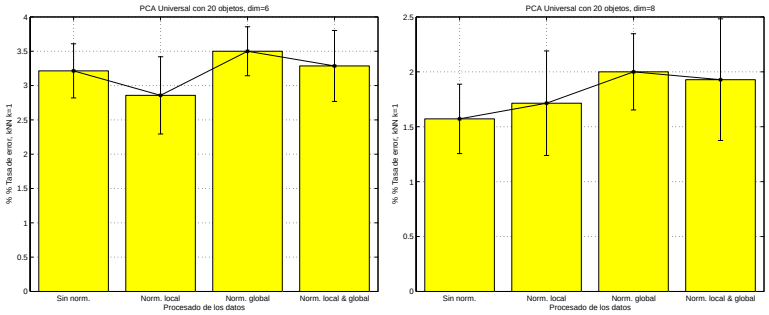
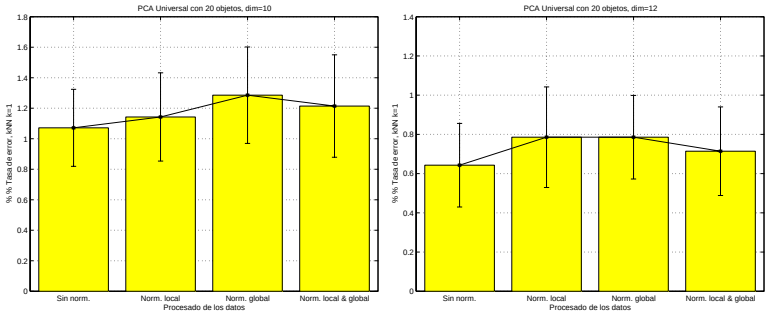


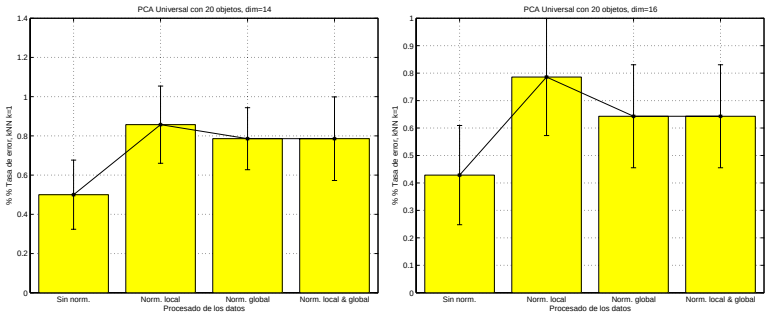
Figura 5.17: PCA Universal con 20 objetos.



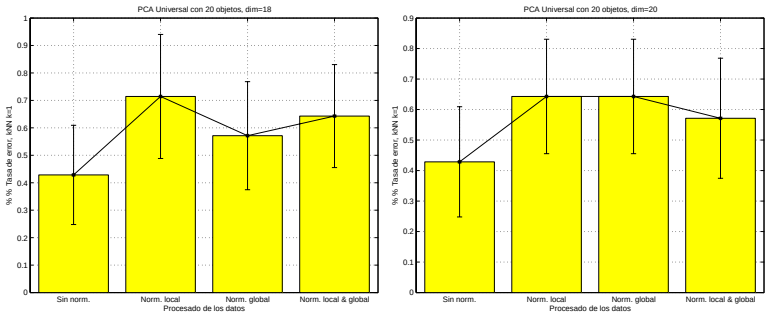
(a) PCA Universal, dim=6 (b) PCA Universal, dim=8



(c) PCA Universal, dim=10 (d) PCA Universal, dim=12



(e) PCA Universal, dim=14 (f) PCA Universal, dim=16



(g) PCA Universal, dim=18 (h) PCA Universal, dim=20

Figura 5.18: PCA Universal, influencia de la normalización de los datos.

La figura 5.18 muestra los mismos resultados con un mayor nivel de detalle: cada gráfico se corresponde con uno de los niveles de compresión empleados. En esta representación se observa con gran claridad cómo las diferencias entre los distintos métodos no son significativas, si bien los mejores resultados se obtienen en casi todos los casos cuando no se realiza normalización de las imágenes.

Dado que las diferencias encontradas no son todo lo relevantes que cabría esperar, se decidió realizar nuevos experimentos, en una situación más parecida a la que podría encontrarse en una aplicación real: introduciendo ruido en las imágenes de test.

Para realizar las pruebas, se consideran dos tipos de ruido distintos. En primer lugar, se utiliza un ruido gaussiano que suma a cada píxel de la imagen un valor con distribución normal de media y varianza ajustables. En los experimentos se utiliza media cero y varianzas comprendidas entre 0.02 (nivel de ruido bajo) y 0.10 (nivel de ruido elevado). Este ruido gaussiano puede simular faltas de uniformidad en la iluminación o presencia de sombras en una aplicación real. Por otra parte, también se realizan experimentos con un ruido tipo salt & pepper (sal y pimienta) que modifica píxels aleatorios de la imagen, sustituyéndolos por puntos negros o blancos. El porcentaje de pixels afectados por ruido es ajustable, y se realizan experimentos en los que este porcentaje varía entre el 4 % (imagen poco deteriorada) y el 20 % (imagen altamente deteriorada). Este tipo de ruido pretende simular el efecto de defectos locales que se podrían producir en la captura de imágenes en una aplicación real. Como ejemplo, la figura 5.19 muestra un ejemplo de imágenes de entrenamiento y test para un ensayo con presencia de ruido. Se trata de imágenes del mismo objeto, y en las imágenes de test se ha introducido un ruido tipo salt & pepper que afecta al 4 % de los pixels.

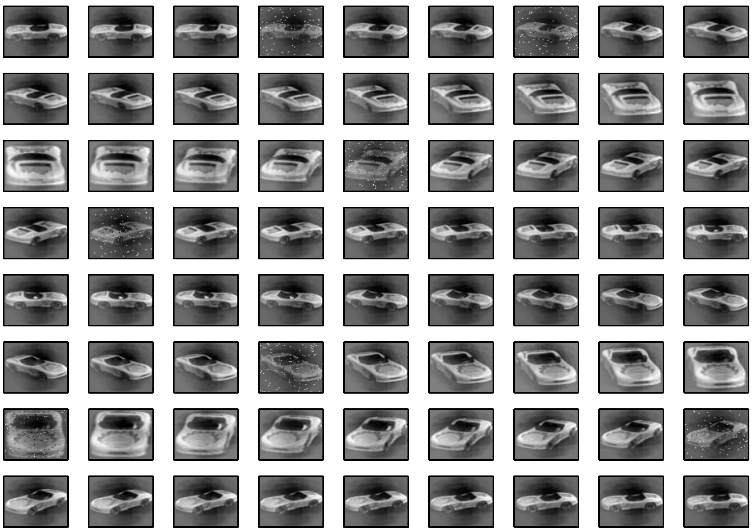


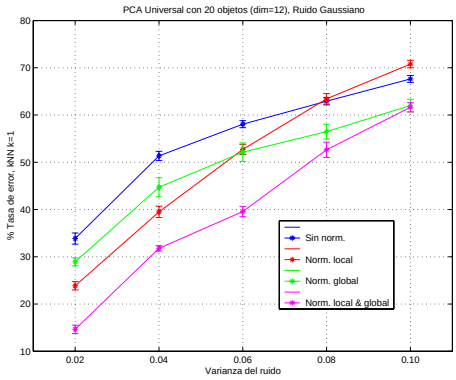
Figura 5.19: Imagenes de entrenamiento y test para el coche, utilizadas en los experimentos de capacidad de reconocimientoo con ruido salt & pepper (ruido sólo en imágenes de test).

Los experimentos se realizan de forma similar al caso sin presencia de ruido: se utiliza la validación cruzada de 10 divisiones para seleccionar aleatoriamente los subconjuntos de entrenamiento y test, y sobre las imágenes del subconjunto de test se introduce el ruido. Se llevan a cabo diez repeticiones del experimento, al igual que en el caso anterior, para conseguir una mayor fiabilidad en los resultados.

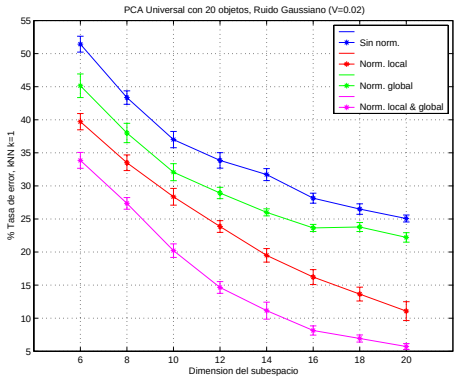
Los resultados obtenidos en cuanto a tasas de error en clasificación con ruido de tipo gaussiano se muestran en la figura 5.20. Se observa un comportamiento más acorde con lo esperado, en el que sí queda patente la influencia de realizar o no una normalización previa de las imágenes. La figura 5.20(a) representa el efecto de introducir un ruido en las imágenes de test con varianza variable entre 0.02 y 0.10 pero con el número de componentes principales utilizado (compresión de la información) fijo e igual a 12. Por supuesto, las tasas de error son mucho mayores que las obtenidas en ausencia de ruido, y llegan a ser muy elevadas para una varianza de 0.10. Se observa cómo la normalización es beneficiosa en todo el rango de niveles de ruido, y cómo las diferencias, en este caso, sí son estadísticamente significativas. La mejor opción es siempre utilizar normalización tanto local como global; y en general los peores resultados se obtienen siempre cuando no se aplica ninguna de las dos normalizaciones. La conclusión es que ambas normalizaciones ayudan a mejorar las tasas de reconocimiento. Las figuras 5.20(b), (c), (d), (e) y (f) muestran resultados para experimentos en los que el nivel de ruido es fijo pero el nivel de compresión utilizado es variable. En estas figuras puede observarse cómo la normalización previa de las imágenes resulta particularmente beneficiosa cuando el número de componentes utilizado es elevado. Por ejemplo, con un ruido de varianza 0.2, la tasa de error media disminuye desde un 25 % hasta menos de un 6 %.

El experimento anterior se repite con ruido tipo salt & pepper. La figura 5.21 muestra los resultados obtenidos. Las conclusiones son similares a las obtenidas con el ruido gaussiano: en general, la mejor opción es la realización de las dos normalizaciones (local y global); y los peores resultados se obtienen con los datos sin normalizar. La figura 5.21(a) muestra cómo la mejora en tasa de reconocimiento se produce en todos los niveles de ruido (si bien para niveles muy elevados la mejora es reducida) y las figuras 5.21(b), (c), (d), (e) y (f) muestran como, para cada intensidad de ruido, la mejora se produce en todos los niveles de compresión. Las diferencias son significativas y elevadas, particularmente cuando el número de componentes es alto.

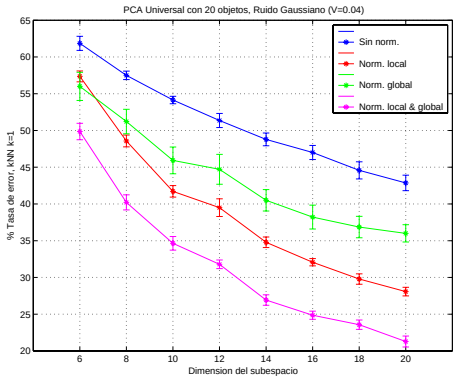
Por último, en la figura 5.22 se muestra también la influencia de la normalización en la arquitectura subespacio objeto modificado.



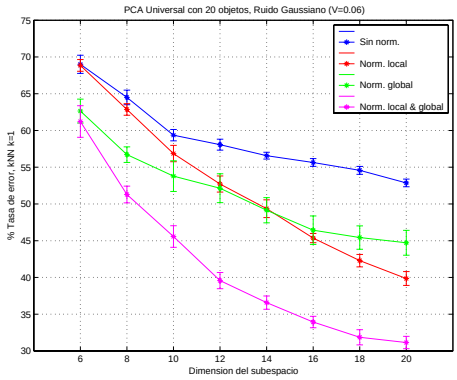
(a) PCA Universal (dim=12), Ruido Gaussiano



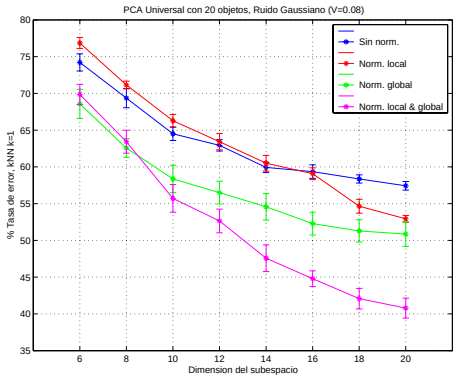
(b) PCA Universal, Ruido Gaussiano V=0.02



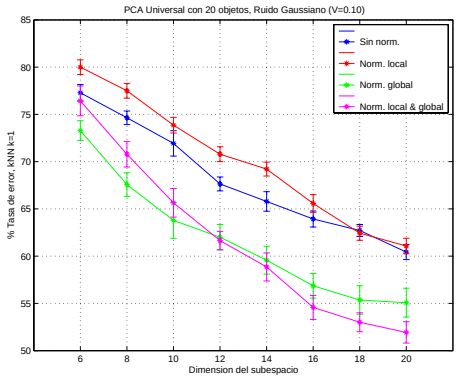
(c) PCA Universal, Ruido Gaussiano V=0.04



(d) PCA Universal, Ruido Gaussiano V=0.06

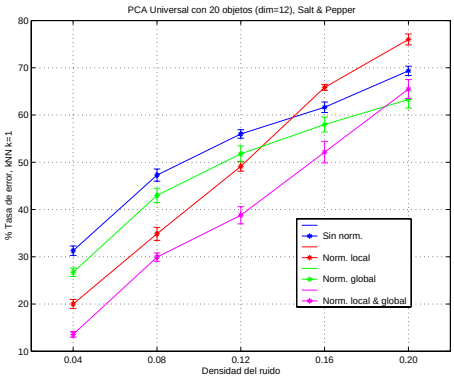


(e) PCA Universal, Ruido Gaussiano V=0.08

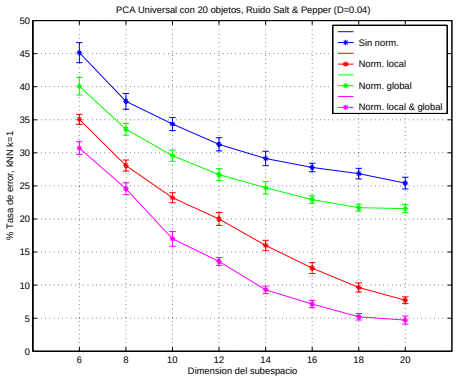


(f) PCA Universal, Ruido Gaussiano V=0.10

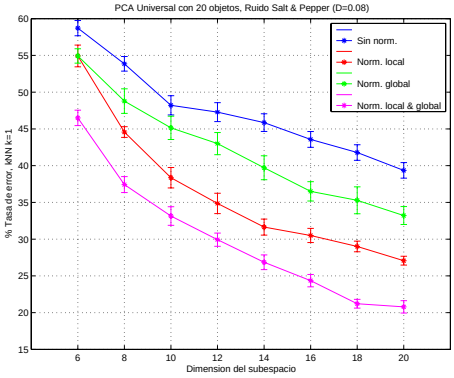
Figura 5.20: PCA Universal, Ruido Gaussiano .



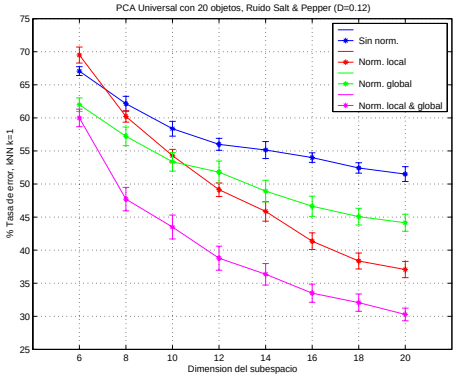
(a) PCA Universal (dim=12), Ruido Salt & Pepper



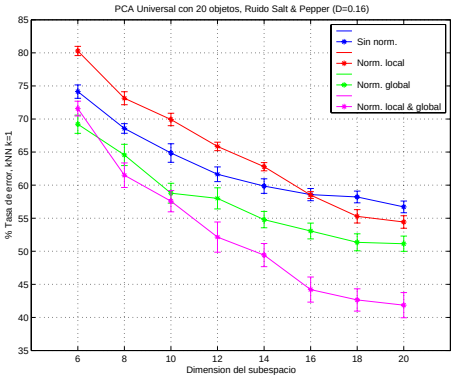
(b) PCA Universal, Ruido Salt & Pepper D=0.04



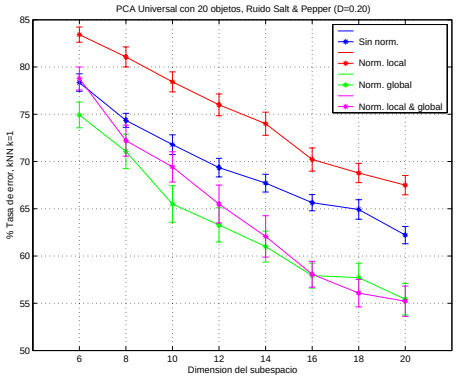
(c) PCA Universal, Ruido Salt & Pepper D=0.08



(d) PCA Universal, Ruido Salt & Pepper D=0.12

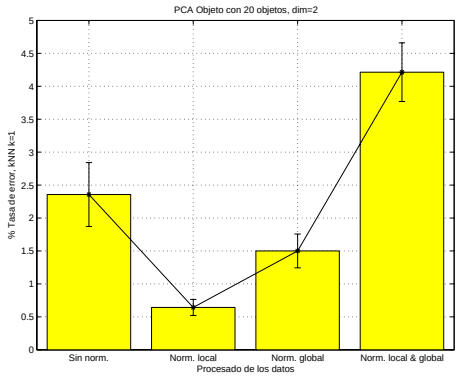


(e) PCA Universal, Ruido Salt & Pepper D=0.16

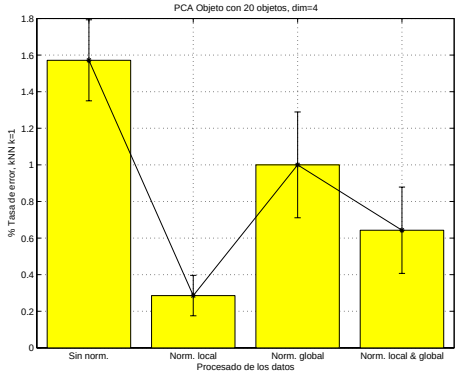


(f) PCA Universal, Ruido Salt & Pepper D=0.20

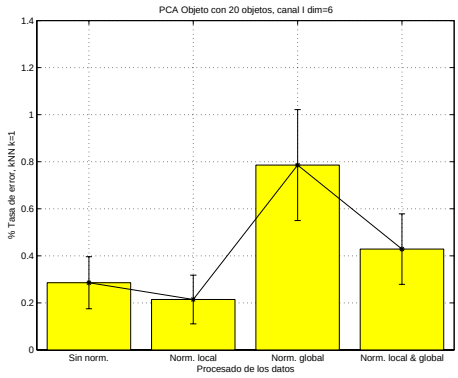
Figura 5.21: PCA Universal, Ruido Salt & Pepper.



(a) PCA Objeto Modificado, canal I, dim=2



(b) PCA Objeto Modificado, canal I, dim=4



(c) PCA Objeto Modificado, canal I, dim=6

Figura 5.22: PCA Objeto Modificado, canal I, influencia de la normalización de los datos.

A partir del análisis sobre el efecto de las normalizaciones, se pueden extraer dos conclusiones: en primer lugar, realizar o no normalización de los datos (sea esta local o global) afecta a las tasas de reconocimiento, y por tanto puede afectar a la comparación entre diferentes algoritmos de extracción de características como PCA e ICA. En segundo lugar, también se concluye que las mejores tasas de éxito se obtienen cuando se normalizan las imágenes local y globalmente: cuando no existe ruido en las imágenes de test (situación ideal) las diferencias son poco significativas; pero cuando sí existe ruido (situación más próxima a la realidad) los porcentajes de clasificaciones correctas se incrementan con la doble normalización.

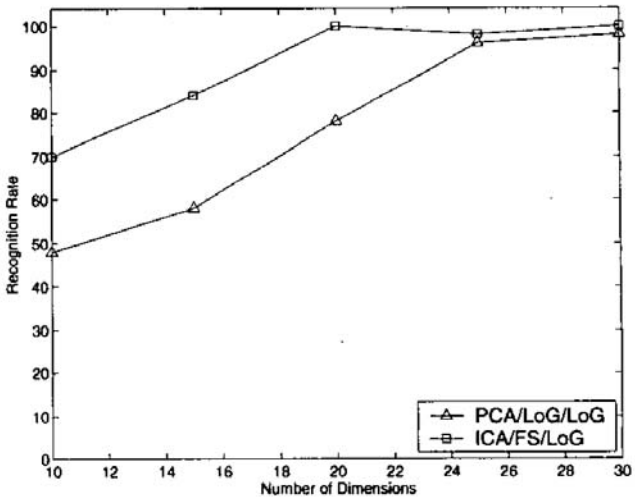


Figure 4. Plot of recognition rate for best ICA and PCA results.

Figura 5.23: Gráfico extraído de Fortuna *et al.* [61], queda patente que en el caso de utilizar todas las dimensiones, los resultados de ambos métodos coinciden.

5.1.3. Conclusiones

En las secciones precedentes se ha demostrado cómo las técnicas de ICA y PCA están íntimamente relacionadas y bajo qué circunstancias su funcionamiento es equivalente. Si no se realiza selección de características, ICA y la operación de *whitening* (PCA + escalado) funcionan de manera totalmente equivalente con cualquier tipo de datos si el clasificador utilizado es invariante rotacional. Sin embargo, si se realiza posteriormente una normalización en el cálculo de las componentes independientes, (como es el caso del algoritmo de Comon [37]), o bien se realiza una selección de características previa al clasificador, o el clasificador utilizado no es invariante frente a rotaciones, el funcionamiento, y por tanto, las tasas de éxito en el reconocimiento de ambos métodos pueden ser significativamente diferentes. Basándonos en dichas valoraciones es posible explicar los aparentes resultados contradictorios de algunos de los trabajos donde se realizan comparativas entre PCA e ICA:

- En el trabajo de Martiriggiano *et al.* [110] se utiliza el algoritmo *FastICA* para el cálculo de las componentes independientes y la distancia euclídea como clasificador, y sin embargo, los resultados en reconocimiento de ICA son superiores a los de PCA. La razón puede hallarse en que no se está realizando el escalado posterior a la decorrelación de los datos en PCA (*whitening*), y por tanto las distancias entre los puntos son diferentes en ambos métodos.

- La capacidad de reconocimiento de caras mediante análisis multiresolución se examina en Ekenel y Sankur [51] mediante la comparativa PCA/ICA utilizando las arquitecturas descritas en esta tesis en la sección 5.1.1.3. Aunque su trabajo afirma el mejor funcionamiento de ICA con la arquitectura II, sus resultados experimentales muestran también que las tasas de reconocimiento de PCA e ICA con la misma arquitectura (la arquitectura I) son totalmente equivalentes cuando se utilizan medidas de distancias invariantes rotacionales: norma L2 y coseno; como corresponde con nuestras conclusiones.

- En el trabajo de Fortuna *et al.* [61] los resultados indican que ICA proporciona mejores tasas de reconocimiento, pero los autores realizan un proceso de selección de características con las componentes independientes. Los resultados de ICA se comparan con los proporcionados por las componentes principales ordenadas según sus valores propios. A pesar de que los autores afirman la superioridad de ICA tanto en este trabajo como en otros posteriores [58][59][60], la única razón de tal superioridad se encuentra en la selección de características que realizan con ICA y no con PCA.

  Tal como se demuestra en esta tesis y se ilustra en el experimento del apartado 5.1.1.5, en el caso de utilizar todas las dimensiones, los resultados de ambos métodos deben coincidir. Esto queda patente en los resultados de los propios autores tal y como se muestra en el grafico de la figura 5.1.2.2, extraido de [61].

- Moghaddam [114] mediante un estudio comparativo experimental de ambos métodos concluye que el funcionamiento de PCA e ICA es equivalente, de acuerdo a nuestras afirmaciones. Sin embargo, Sahambi y Khorasani [144], aunque aportan resultados de ICA ligeramente superiores a los de PCA concluyen que ambas técnicas son muy dependientes del tipo de datos utilizados. La razón de que sus resultados de ICA difieran de los de PCA puede hallarse en la omisión del escalado posterior a la decorrelación de los datos en PCA (*whitening*).

- En Bartlett *et al.* [11] la diferencia de resultados de PCA/ICA es debida a que se utilizan subconjunto de componentes, es decir, se realiza un proceso de selección de características.

- En el trabajo de Draper *et al.* [47] se muestran multitud de resultados comparativos de ambas técnicas, utilizando diversas medidas de distancia como clasificadores (entre ellas, la norma L2 y el coseno) y bajo las dos arquitecturas. En general, las tasas de éxito en el reconocimiento de caras aportadas para ICA con la arquitectura II y utilizando distancia euclídea son semejantes a los resultados de PCA con la distancia de Mahalanobis (equivalente a *whitening* PCA con distancia euclídea), por tanto sus resultados son coherentes con nuestras afirmaciones.

- Finalmente, tal como hemos expuesto, ICA y PCA pueden funcionar de manera diferente si se realiza un paso de normalización posteriormente a la obtención de las componentes independientes (*e.g.* empleando el algoritmo de Comon [37]), tal como se muestra en los trabajos de Liu *et al.* [101] [102] [104] .

5.2. RP vs. PCA

En la seccion 3.5 se expuso el método de proyecciones aleatorias o RP como procedimiento de reducción de dimensionalidad de los datos. Normalmente, RP se utiliza como paso previo al cálculo de componentes principales (PCA) cuando los datos presentan dimensiones muy elevadas. En muy pocas ocasiones se utiliza RP directamente como método de extracción de características, y no se han encontrado antecedentes de su utilización en aplicaciones de reconocimiento de objetos.

Como aportación de la presente tesis, se estudia la aplicabilidad del método RP como extractor de características para reconocimiento de objetos. Los apartados posteriores muestran tanto las ventajas teóricas esperables como los resultados experimentales obtenidos, siempre comparados frente a PCA como método clásico de extracción de características.

5.2.1. Ventajas de aplicar RP al reconocimiento de objetos

Existen diversas razones que aconsejan el uso de la técnica de RP como método de extracción de características en el reconocimiento de objetos basado en la apariencia. La primera de ellas es que la matriz de transformación **R** definida en la ecuación 3.74 es independiente de los datos, a diferencia del resto de técnicas de extracción de características; puesto que tanto los métodos supervisados (como LDA) como los no supervisados (PCA, ICA y NMF), utilizan un conjunto de datos de entrenamiento para el cálculo de los coeficientes de la matriz de transformación. Esto implica que la matriz de transformación de RP es mucho más rápida de calcular que la de cualquier otro método.

Esta diferencia en tiempo de cómputo ha de ser puntualizada: el tiempo requerido para la clasificación de un objeto con cualquiera de los métodos de extracción de características considerados a lo largo de esta tesis es similar, dado que en todos los casos la operación a realizar es una proyección (un producto matricial). Sin embargo, el tiempo requerido para obtener la matriz de proyección (tiempo de cómputo *off-line*) es muy distinto en función de los métodos y, en general, aumenta considerablemente cuando la dimensión de los datos es muy elevada. En el caso de PCA (y de aquellas técnicas que utilizan PCA en el preprocesado previo de los datos -como LDA e ICA-), aunque existen algoritmos eficientes para el cómputo de los primeros vectores propios sin la necesidad de utilizar la matriz de covarianzas original, el número de vectores está limitado por el número de ejemplos de entrenamiento [5]. Lo que implica que estos algoritmos pueden no ser muy útiles cuando se disponen de conjuntos de datos muy reducidos. Sin embargo,en el caso del método RP, este tiempo *off-line* es extremadamente reducido, dado que **W** se obtiene

[5] Recordemos que en el caso de trabajar con un conjunto de N imágenes de dimensión M, para el cálculo de la matriz de transformación de PCA, la dimensión relevante es la menor de ambas (truco de Turk y Pentland[157]).

aleatoriamente.

Para ilustrar esta ventaja importante de RP respecto del resto de los métodos, la figura 5.24 muestra sobre una escala logarítmica la diferencia en tiempo de cómputo *off-line* entre los distintos métodos. Los valores se representan como $log_{10}\frac{t}{t_{RP}}$, donde t representa el tiempo de cómputo de cada uno de los métodos y t_{RP}representa el tiempo de cómputo del método RP, que se toma como referencia. Los resultados muestran que PCA requiere un tiempo 3 órdenes de magnitud superior al de RP, y tanto ICA como NMF un tiempo 6 órdenes de magnitud superior.

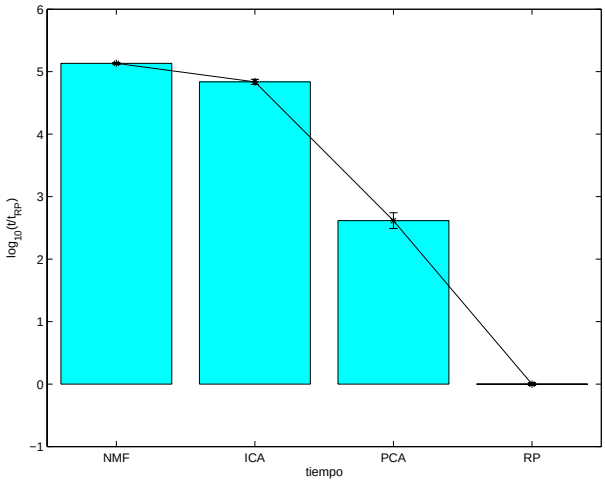


Figura 5.24: Tiempos de cómputo *off-line* de distintos métodos de extracción de características.

Calcular una matriz aleatoria de dimensiones $K \times M$ utilizando el algoritmo presentado en el apartado 3.5 tiene una carga computacional $O(MK^2)$, y si se prescinde del paso de la ortonormalización, de sólo $O(MK)$ [5]. Por otro lado, PCA tiene un coste computacional de $O(M^2N) + O(M^3)$ [88]. Cabe mencionar que el coste computacional de la proyeccion (el producto matricial) es de $O(MKN)$ para cualquier método. Sin embargo, si la matriz de la transformación se aproxima utilizando valores enteros como se describe en [5], las proyecciones pueden calcularse más rápidamente.

Otra gran ventaja de RP frente al resto de los métodos debido a su independencia de los datos, es que en el caso de que los patrones sean variables en el tiempo, la matriz de la transformación no debe ser recalculada continuamente. Mientras que en el resto de métodos, si los patrones cambian, es necesario el cálculo de la matriz de transformación y la consiguiente proyección de los datos de entrenamiento para asegurar el mantenimiento óptimo de sus propiedades. Por tanto, las aplicaciones que más se pueden beneficiar del

uso de la técnica de RP son aquéllas en las que la matriz de proyección debe ser recalculada continuamente (la base de datos de objetos a reconocer varía: aparecen o desaparecen nuevos elementos durante el funcionamiento). Ejemplos de tales aplicaciones se pueden encontrar en controles de acceso en los que se desea mantener un registro de las personas que permanecen en el interior de cierto edificio, o en aplicaciones de índole similar.

Los resultados experimentales que a continución se presentan, indican que la técnica de RP produce unos resultados de reconocimiento algo inferiores a los de PCA, pero comparables a éstos aún tratándose de una técnica tan sencilla. Por lo que su uso en entornos como los descritos antes es del todo recomendable.

5.2.2. Resultados experimentales

En esta sección se demuestra experimentalmente la validez de utilizar RP como técnica de reducción de la dimensionalidad en los sistemas de reconocimiento de objetos basados en la apariencia. En concreto, se presentan los resultados experimentales obtenidos tras comparar las técnicas de PCA y RP[6]. Se han utilizado dos librerías para la realización de este test:

- Un subconjunto de 20 objetos de la librería COIL-100: los 20 objetos más similares entre sí en apariencia y por tanto, los más complejos de clasificar correctamente. Esta selección se ha realizado para evitar tasas de éxito en el reconocimiento demasiado elevadas que puedan oscurecer los resultados de la comparativa. La figura 5.25 muestra estos 20 objetos más complejos de la COIL-100.
- Un subconjunto de 30 sujetos de la librería de caras humanas ORL. La figura 3.12 muestra una imagen de cada unos de los 30 individuos.

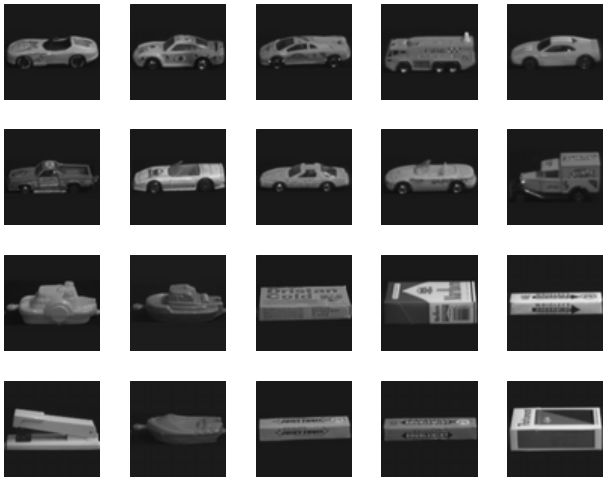
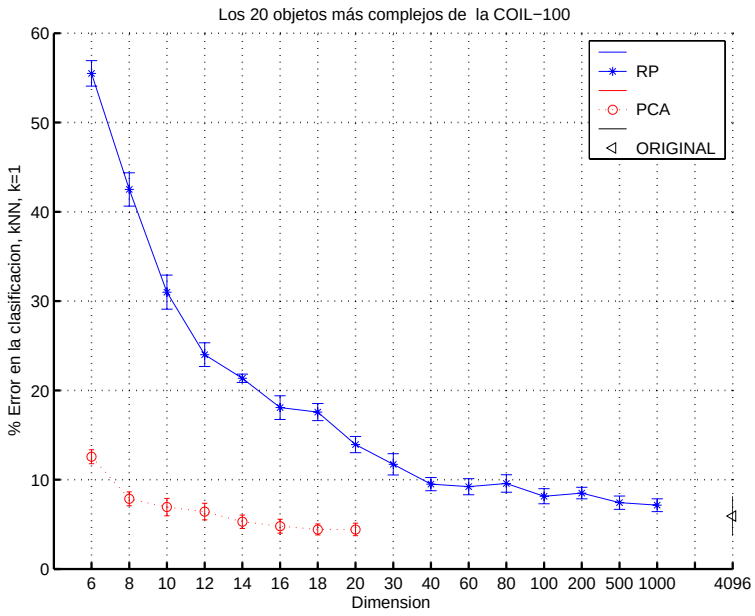
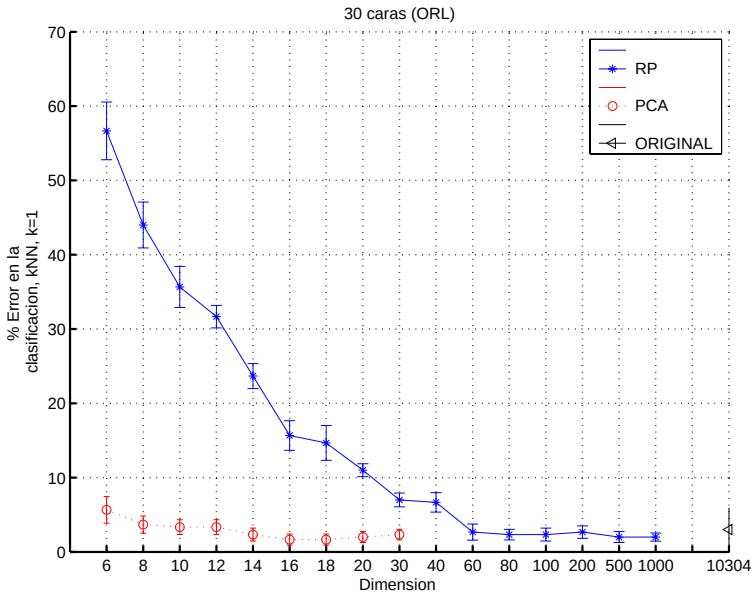


Figura 5.25: Un subconjunto de 20 objetos de la librería COIL-100: los 20 objetos más similares entre sí en apariencia y por tanto, los más complejos de clasificar correctamente.

[6]Parte de estos resultados se pueden consultar en [166] y [167].



(a) Objetos (librería COIL-100)



(b) Caras (librería ORL)

Figura 5.26: Tiempos de cómputo *off-line* de distintos métodos de extracción de características.

Para testear el funcionamiento de RP, el clasificador utilizado ha sido el vecino más cercano con distancia euclídea (norma L2) junto con los siguientes datos de entrada:

- Las imágenes en el canal de apariencia I sin ningún tipo de compresión, esto es, los valores en niveles de gris de las imágenes es el espacio imagen original(de tamaño 64×64 para los objetos de la COIL-100, y de 112×92 para las caras de la ORL).
- Las proyecciones en el subespacio PCA universal obtenido a partir de las imágenes originales en niveles de gris: se han realizado diversos test variando el número de componentes del subespacio utilizado, desde 6 hasta el número máximo de vectores propios (igual o inferior al número de imágenes de entrenamiento utilizadas)
- Las proyecciones en el subespacio RP universal: se han realizado diversos test variando el número de dimensiones de la matriz de proyecciones aleatorias, desde 6 hasta 1000.

En todos los casos, se ha realizado un proceso de validación cruzada mediante 10 repeticiones para obtener resultados más fiables. Estos resultados se presentan en las figuras 5.26.a y 5.26.b. Es posible observar que ambas técnicas proporcionan resultados muy similares a los obtenidos por el clasificador utilizando puntos del espacio imagen, lo que indica que ambas técnicas son válidas para la compresión de la información del espacio original. Sin embargo, RP requiere un mayor número de características que PCA para obtener los mismos resultados de clasificación y particularmente, con los objetos de la librería COIL-100, los resultados de PCA son ligeramente superiores a los mejores resultados obtenidos con RP. El número de características requeridas para una tasa de éxito considerable al utilizar RP, aún siendo superior a la necesitada por PCA, es muy inferior al número de píxeles de la imagen en el espacio original, por lo que la reducción de la dimensión se consigue y el clasificador gana en rapidez.

A modo de conclusión, cabe decir que PCA debe utilizarse cuando el tiempo de cómputo *off-line* no es relevante, puesto que consigue una reducción de la dimensión mayor que RP para tasas de éxito en el reconocimiento equivalentes; y RP debe utilizarse cuando el tiempo de cómputo *off-line* es importate, i.e. cuando el número de patrones (y de imágenes de entrenamiento) no es fijo, y el entrenamiento debe realizarse contínuamente. En tales casos, es suficiente incrementar el número de características con el objetivo de conseguir resultados similares a los proporcionados por PCA, pero evitando el alto coste computacional requerido para el cálculo de las componentes principales.

5.3. NMFSC vs. PCA

La característica fundamental de la descomposición NMF sobre imágenes es su capacidad para producir representaciones locales de los objetos como se mostró en el ejemplo de la figura 3.15(b) dentro del apartado 3.6 de la presente tesis. Para obtener este tipo de representación, las imágenes generadoras del subespacio deben ser de la misma clase, por ejemplo, una base de datos formada por caras o por vistas del mismo objeto. Por ello, el subespacio a utilizar ha de ser de tipo objeto, dado que si se utilizase un subespacio universal, la matriz de datos $\mathbf{X}$ contendría imágenes de distintos objetos que, por tanto, no serían similares. Esta es una restricción del método NMF: es necesario utilizar subespacios objeto si se desea obtener representaciones locales. Sin embargo, el algoritmo original de Lee and Seung [96] no genera ese tipo de representaciones por igual con todos los conjuntos de imágenes. Como ya se comentó en la sección 3.6, recientemente Hoyer [75] ha presentado un nuevo algoritmo de NMF, denominado *Non-negative matrix factorization with sparseness constraints (NMFSC)* que incorpora el control sobre la restricción de *sparseness* para mejorar las descomposiciones de las imágenes. Utilizando este nuevo algoritmo aplicado sobre las imágenes de la librería COIL-100 se han obtenido vectores en los que sí se aprecian características locales de los objetos. La figura 3.15(b) muestra un ejemplo de ello con el objeto 29.

En este apartado, se pretende realizar una comparativa del funcionamiento de PCA frente al de NMFSC dentro del ámbito del reconocimiento de objetos basado en su apariencia visual. Pero antes de mostrar y discutir los resultados obtenidos tras la realización de dicho estudio, se hace hincapié en la forma más idónea de utilizar NMFSC al trabajar con imágenes como datos de entrada.

5.3.1. Preprocesado de los datos

Los vectores proporcionados por los algoritmos de NMF (el algoritmo original de Lee y Seung [96] o bien la nueva versión de Hoyer [75]), $\mathbf{W}$, provienen de la matriz de datos $\mathbf{X}$ y dada la naturaleza de la transformación las componentes de $\mathbf{X}$ deben ser números positivos o nulos. Luego, a primera vista, parece evidente que la técnica de NMF podría aplicarse directamente sobre imágenes en niveles de gris puesto que sus valores son intrínsecamente positivos (0-255). Sin embargo, ciertos preprocesamientos de las imágenes pueden beneficiar la transformación NMF produciendo vectores de características donde las características locales son más notorias. Por ejemplo, el preprocesado descrito a continuación (mediante pseudocódigo) fue propuesto por Lee y Seung [96] para tratar las imágenes de las caras antes de aplicar la transformación NMF:

$$\begin{aligned}
\mathbf{X} &:= \mathbf{X} - media(\mathbf{X}) \\
\mathbf{X} &:= \mathbf{X} / \sqrt{media(\mathbf{X})^2} \\
\mathbf{X} &:= \mathbf{X} + 0{,}25 \\
\mathbf{X} &:= \mathbf{X} * 0{,}25 \\
\mathbf{X} &:= \text{mín}(\mathbf{X}, 1) \\
\mathbf{X} &:= \text{máx}(\mathbf{X}, 0)
\end{aligned} \tag{5.15}$$

mientras que Li [100], propone el siguiente preprocesado de datos:

$$\begin{aligned}
\mathbf{X} &:= \mathbf{X} - \text{mín}(\mathbf{X}) \\
\mathbf{X} &:= \mathbf{X} * 255 / \text{máx}(\mathbf{X})
\end{aligned} \tag{5.16}$$

Adicionalmente al uso de alguno de estos dos procesados, es necesario hacer la siguiente operación:

$$\mathbf{X} := \text{máx}(\mathbf{X}, 10^{-4}) \tag{5.17}$$

para evitar tener ceros en los datos, puesto que la operación de divergencia que incluye el algoritmo de NMF no puede manejarse con ceros exactos [75][96].

Además de estos preprocesados, que al fin y al cabo son normalizaciones de los datos dentro de un rango de valores positivos, Hoyer [75] propone el empleo de un nuevo preprocesado para el tratamiento de los datos procedentes de las imágenes naturales, puesto que otra de las aplicaciones de NMF, paralelamente a ICA, es su uso en Neurociencia para la generación de modelos matemáticos de los campos receptivos de las células neuronales de la corteza visual primaria. Y que en esta tesis lo haremos extensivo al caso de utilizar imágenes de objetos reales. Este nuevo preprocesado, de inspiración biológica y denominado *canal ON-OFF*, se basa en un procesado similar al que ocurre con la información procedente de la retina antes de llegar a la corteza visual .

Hemos querido aplicar este preprocesado a las imágenes de los objetos, con el fin de mejorar la obtención de las características locales proporcionadas por NMFSC, y tanto cualitativamente, mediante la visualización de los vectores de NMFSC, como cuantitativamente, mediante las tasas de éxito en el reconocimiento obtenidas experimentalmente, el empleo del canal ON-OFF proporciona un aumento de las características locales presentes en los vectores NMFSC, así como también se incrementan las tasas de éxito en el reconocimiento, al utilizar NMFSC junto con el canal ON-OFF para comprimir la información de las imágenes de los objetos y generar su modelo de apariencia.

El canal o preprocesado ON-OFF está basado en el campo receptivo de las neuronas de la zona LGN [41] y se puede expresar mediante el siguiente pseudocódigo:

$$\begin{aligned} \mathbf{Xp} &:= \text{máx}(\mathbf{X}, 0) \\ \mathbf{Xn} &:= -\text{mín}(\mathbf{X}, 0) \\ \mathbf{X}_{ON-OFF} &:= [\mathbf{Xp}; \mathbf{Xn}] \end{aligned} \tag{5.18}$$

donde se cumple

$$\mathbf{X} = \mathbf{Xp} - \mathbf{Xn} \tag{5.19}$$

De las ecuaciones anteriores se puede apreciar que el canal ON-OFF es, básicamente, la concatenación de la doble información contenida en una imagen ($\mathbf{x}$) o un conjunto de éstas ($\mathbf{X}$)[7], mediante la obtención de los valores positivos por un lado ($\mathbf{Xp}$) y los negativos (convertidos a valores positivos) por otro ($\mathbf{Xn}$). El filtrado a través del canal ON-OFF se realiza a partir de imágenes en niveles de gris (o bien una transformación de ellas). Por lo tanto, las imágenes en el canal ON-OFF son vectores de tamaño doble a su tamaño original en niveles de gris.

El preprocesado de las imágenes a través del canal ON-OFF puede realizarse a partir de las imágenes en niveles de gris de los objetos, pero también a partir de otras representaciones de la imagen. Nuestra propuesta es utilizar el canal ON-OFF sobre las imágenes de bordes de los objetos, basándonos en el proceso o camino que sigue la información visual en el cerebro de los mamíferos [40].

Para realizar la detección de bordes de las imágenes de los objetos se propone utilizar operadores de segunda derivada, por ser transformaciones más naturales[8] que las de otros operadores más complejos como por ejemplo, el detector de Canny[31]. Dentro de los operadores de segundo orden, destacan por su simplicidad, los operadores *laplaciana (L)* y *laplaciana de la gaussiana (LoG)*. El operador L es un operador de segunda derivada definido como :

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{5.20}$$

donde I representa la imagen bidimensional $I(x, y)$. Una posible aproximación del operador L para su utilización en el procesado de las imágenes es la siguiente máscara que presenta simetría rotacional (lo que significa que permite detectar bordes en todas las direcciones espaciales):

$$L = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{5.21}$$

[7] Se recuerda que a lo largo de esta tesis se viene utilizando $\mathbf{X}$ para denotar la matriz de datos, donde cada uno de sus vectores columna, $\mathbf{x}$, es una imagen.

[8] Más similares a las funciones matemáticas utilizadas para modelar los procesos de transmisión de información en el cerebro.

El operador LoG se define como:

$$\nabla^2(G \otimes I) = \nabla^2 G \otimes I \tag{5.22}$$

donde G es la función gaussiana:

$$G(x, y) = -\frac{1}{\sqrt{2\pi}\sigma} e^{-(x^2+y^2)/2\sigma^2} \tag{5.23}$$

El operador LoG además ser simétrico rotacional permitiendo la detección de bordes isótropa, es un operador capaz de trabajar a diferentes escalas con la simple variación de la desviación standard σ. Una máscara que aproxima al operador LoG para $\sigma = 1{,}4$ es la siguiente:

$$LoG = \begin{pmatrix} 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -23 & -40 & -23 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \end{pmatrix} \tag{5.24}$$

(a) L (b) LoG

(c) L_{ON} (d) LoG_{ON}

(e) L_{OFF} (f) LoG_{OFF}

Figura 5.27: Imágenes en los canales ON-OFF obtenidas a partir de la imagen *Lenna* transformada por los operadores de segundo orden L y LoG.

La figura 5.27 ilustra el efecto de estos dos operadores de segundo orden sobre la popular imagen de *Lenna* junto con el procesado del canal ON-OFF aplicado sobre las imágenes de bordes que proporcionan ambos operadores. Aparentemente, en dicha figura no se distingue, debido a la visualización utilizada, la diferencia de información entre las imágenes en los canales L y LoG y los canales ON respectivos, por lo que la figura 5.28 muestra la diferencia entre los canales ON-OFF y los de origen (L y LoG) para una mejor comprensión del efecto de estos canales sobre las imágenes de bordes.

(a) $L - L_{ON}$ (b) $LoG - LoG_{ON}$

(c) $L - L_{OFF}$ (d) $LoG - LoG_{OFF}$

Figura 5.28: Imágenes de las diferencias entre los canales ON-OFF y sus fuentes, L y LoG.

En resumen, para la obtención de los vectores de NMFSC se propone el empleo del canal ON-OFF junto con la detección de bordes mediante los operadores de segundo orden descritos. La figura 3.15(b) mostraba algunos de estos vectores obtenidos utilizando el preprocesado propuesto, en concreto, utilizando el operador L. Los vectores de características que se obtienen son de doble tamaño puesto que la entrada al algoritmo son las imágenes ON-OFF, pero es posible su visualización realizando la operación de la ecuación 5.19 sobre los vectores:

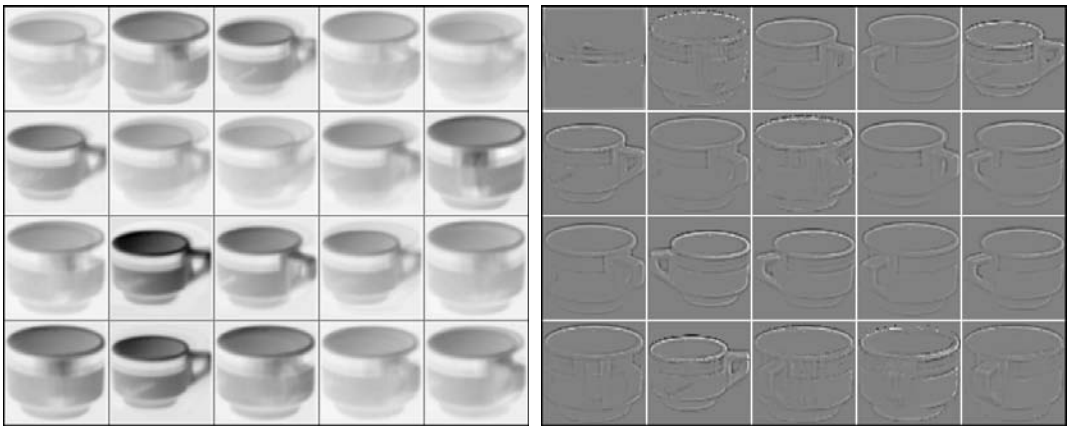
$$\mathbf{W} = \mathbf{W_{ON}} - \mathbf{W_{OFF}} \tag{5.25}$$

En la figura 5.29 se muestran algunos de los vectores de NMFSC obtenidos para el objeto nº 10 de la COIL-100, utilizando el mismo nivel de *sparseness*, $S_W = 0{,}7$, bajo procesados distintos de las imágenes generadoras del subespacio. Si se utilizan las imágenes en niveles de gris filtradas con el canal ON-OFF no se aprecian ningún tipo de características locales del objeto en los vectores de NMFSC obtenidos (figura 5.29(a)); sin embargo, las imágenes de bordes (operador laplaciano) filtradas con el canal ON-OFF producen que los vectores obtenidos muestren ciertos rasgos locales de los objetos. En la figura 5.29(b) es posible apreciar en la imagen superior izquierda como sólo aparece una franja característica de la taza; a su lado izquierdo y en la misma fila superior aparece la imagen de la taza con su asa difuminada, lo más sobresaliente de la imagen es la forma de la taza, este mismo efecto tambien se aprecia en la imagen central de la segunda fila y en dos de las imágenes de la fila inferior.

En la figura 5.30 aparecen algunos de los vectores de NMFSC obtenidos para los objeto nº 7 y nº 81 de la COIL-100, utilizando diferentes niveles de *sparseness*, $S_W = 0{,}4, 0{,}7$ y $0{,}75$, partiendo de las imágenes de bordes (operador laplaciano). Es posible apreciar como para el valor de *sparseness* menor, $S_W = 0{,}4$, no se aprecian características locales en las imágenes de los objetos, y sin embargo, con los otros valores de *sparseness* mayores, $S_W = 0{,}7$ y $0{,}75$ se obtienen imágenes donde se realzan pequeños detalles localizados de los objetos[9].

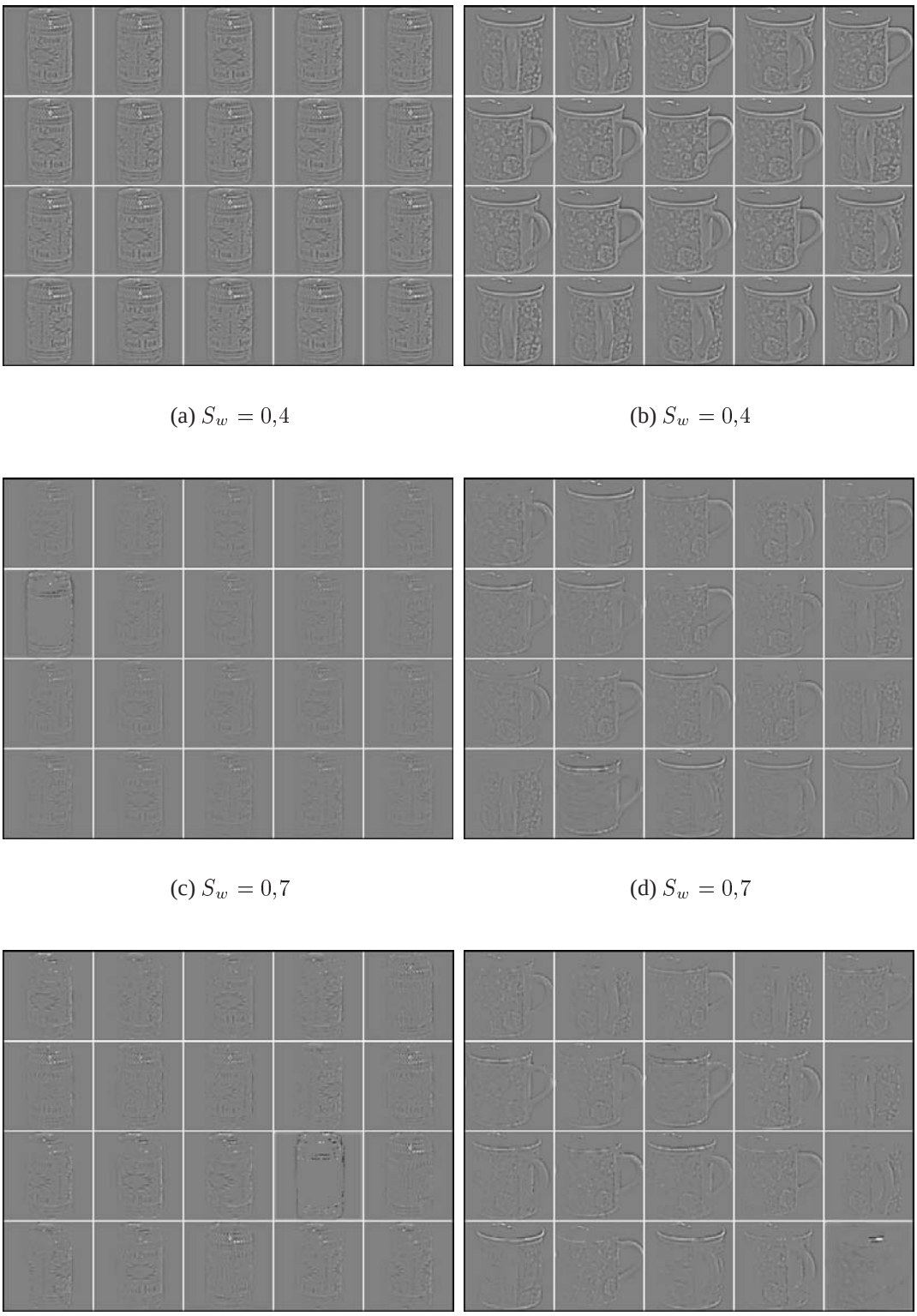
Los experimentos de reconocimiento de objetos que se muestran en el siguiente apartado validan el uso de este preprocesado de las imágenes, pues dan lugar a que la técnica de NMFSC junto con la arquitectura de subespacio objeto modificado, proporcionen unas tasas de aciertos en el reconocimiento de objetos mayores que las conseguidas con la técnica clásica de PCA.

[9]Si debido a la impresión de los gráficos, no es posible distinguir con claridad las características locales en las figuras relativas a los vectores de NMFSC, se recomienda consultar dichas figuras en el formato electrónico de la presente tesis, que se adjunta dentro del CD al final de este documento.



(a) I_{ON-OFF} (b) L_{ON-OFF}

Figura 5.29: Vectores de NMFSC, $S_w = 0{,}7 S_h = 0$, obtenidos de un mismo conjunto de imágenes (objeto nº 10 de la COIL-100) preprocesadas mediante el canal ON-OFF sobre las imágenes en niveles de grises (a); y mediante el canal ON-OFF sobre las imágenes filtradas por el operador laplaciano (b).



(a) $S_w = 0{,}4$ (b) $S_w = 0{,}4$

(c) $S_w = 0{,}7$ (d) $S_w = 0{,}7$

(e) $S_w = 0{,}75$ (f) $S_w = 0{,}75$

Figura 5.30: Vectores de NMFSC para los objetos 7 y 81 de la COIL-100, con tres grados diferentes de *sparseness*, $S_W = 0{,}4, 0{,}7$ y $0{,}75$, partiendo de las imágenes de bordes (operador laplaciano).

5.3.2. Estudio del funcionamiento de NMFSC utilizando la arquitectura de subespacio objeto modificado

Todos los resultados experimentales que se presentan en este apartado, se han realizado utilizando los objetos de la librería COIL-100, en concreto, sus primeros 20 objetos [10]. La arquitectura utilizada para generar los modelos de apariencia es la propuesta en esta tesis, (arquitectura subespacio objeto modificado), y que se ha descrito previamente en la sección 4.2. Se han utilizado el $50\,\%$ de las imágenes de cada objeto para la generación de los subespacios (obtención de los vectores) y el entrenamiento del clasificador, y el otro $50\,\%$ para test. El clasificador utilizado es el vecino más cercano (1-NN) con norma L2, mediante un proceso de validación cruzada de 10 iteraciones, bajo el entorno WEKA [82].

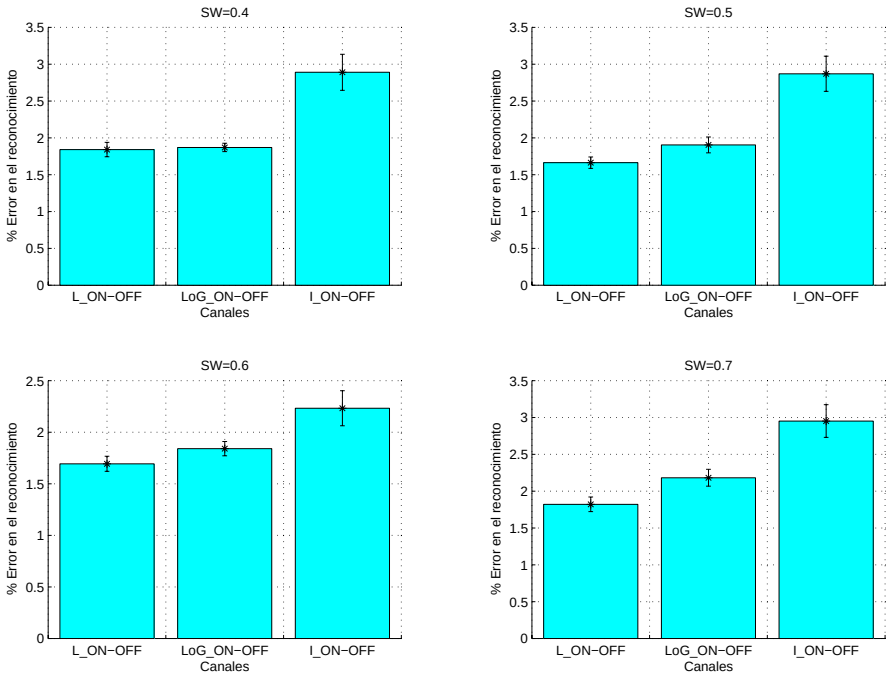
Los resultados obtenidos muestran que el uso de NMFSC junto con el preprocesado antes propuesto permite obtener unas tasas de éxito en el reconocimiento superiores a las de PCA, sobretodo en presencia de oclusiones en las imágenes de test.

5.3.2.1. Comparativa de los canales propuestos

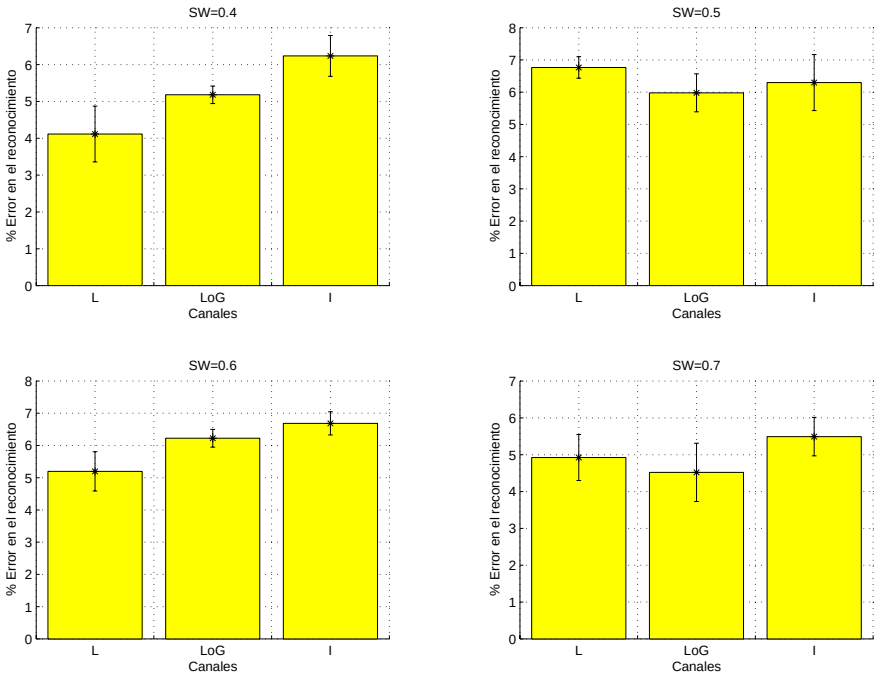
En este apartado, se muestran los resultados obtenidos al realizar una comparativa del funcionamiento de los canales o filtrados propuestos, para las imágenes de los objetos, si se emplea la técnica NMFSC como método de generación de los modelos de apariencia de los objetos.

La figura 5.31 recopila las tasas de éxito obtenidas al realizar la comparativa del funcionamiento de los canales propuestos: L_{ON-OFF} y LoG_{ON-OFF}, frente al uso de los canales I_{ON-OFF} e I, L y LoG, directamente, sin realizar la operación ON-OFF (definida en la ecuación 5.18). La figura 5.31(a) muestra los errores en el reconocimiento obtenidos para las imágenes de test utilizando el preprocesado del canal ON-OFF, y la figura 5.31(b) los errores relativos a los preprocesados sin el empleo del canal ON-OFF. Dichas figuras muestran a su vez, en cuatro subfiguras, los experimentos realizados para cuatro valores de *sparseness* diferentes, $S_w = 0{,}4, 0{,}5, 0{,}6$ y $0{,}7$, y en donde cada punto es la media de 5 experimentos de validación cruzada para 5 dimensiones de subespacios objeto modificado (2, 4, 6, 10 , 20, 30). Al estar el sistema compuesto por 20 objetos, esto equivale a que el numero de componentes de cada subespacio es 40, 80, 120, 200, 400 y 600, respectivamente. La figura 5.32 muestra el resultado de estos 5 experimentos para los canales ON-OFF con $S_w = 0{,}4$ y a partir de esos valores se obtiene el primer subgráfico de la figura 5.31(a).

[10] Se utilizaron sólo los primeros 20 objetos de la librería COIL-100 para disminuir el tiempo de cómputo off-line de la generación de los subespacios objeto.

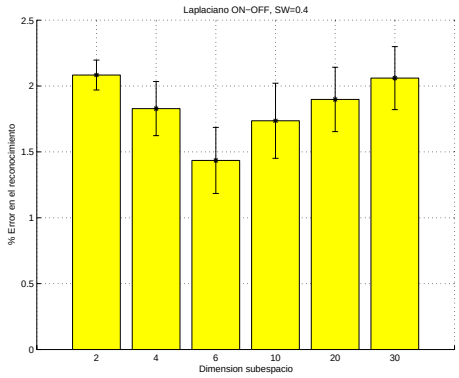


(a) Canales ON-OFF

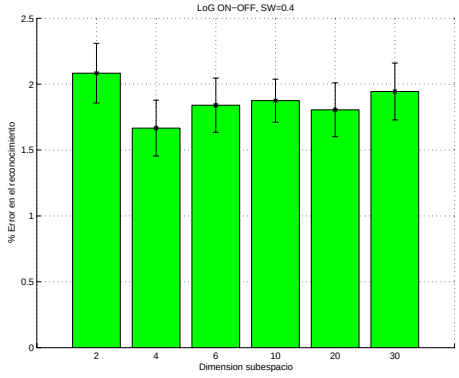


(b) Canales directos

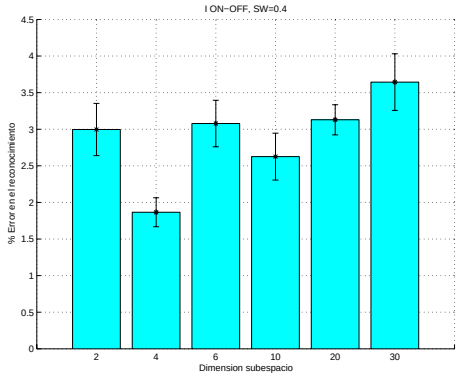
Figura 5.31: Comparativa del funcionamiento de los canales propuestos con NMFSC.



(a) L_{ON-OFF}, $S_w = 0{,}4$



(b) LoG_{ON-OFF}, $S_w = 0{,}4$



(c) I_{ON-OFF}, $S_w = 0{,}4$

Figura 5.32: Comparativa con los canales ON-OFF para $S_w = 0{,}4$.

Los resultados experimentales muestran que los errores de reconocimiento obtenidos para los dos canales propuestos, L_{ON-OFF} y LoG_{ON-OFF}, son inferiores al resto de preprocesados, por lo que en principio resulta interesante utilizar las imágenes de bordes procesadas con el canal ON-OFF antes de emplear NMFSC. En el siguiente apartado, se presenta con más detalle el estudio realizado del funcionamiento de NMFSC, utilizando la arquitectura de subespacio objeto modificado con estos dos canales, frente a la técnica PCA y al uso de dicha arquitectura sin ningún tipo de compresión, es decir, utilizando las imágenes en el espacio original.

5.3.2.2. Comparativa PCA vs. NMFSC

Se ha realizado una comparativa del funcionamiento en reconocimiento de estos dos algoritmos de extracción de características, utilizando subespacios objeto modificado, con un clasificador k-NN con k=1 y norma L2, mediante un proceso de validación cruzada de 10 repeticiones, utilizando, como se ha mencionado antes, el $50\,\%$ de las imágenes de cada objeto para la generación de los subespacios (obtención de los vectores) y el entrenamiento del clasificador, y el otro $50\,\%$ para test.

La figura 5.33 recopila los resultados más relevantes de esta comparativa para el canal de características L_{ON-OFF}. Se han utilizado las siglas O y P, para designar los métodos *original* y PCA, respectivamente, siendo las otras etiquetas (0, 0.4, 0.5, 0.6 y 0.7) los valores de *sparseness* utilizados con el método NMFSC. Los gráficos de la figura 5.34 reproducen los mismos resultados que los mostrados en la figura 5.33 pero por separado para cada uno de los métodos de extracción de características.

Como método espacio imagen *original*, nos referimos sencillamente a utilizar un subconjunto de K imágenes originales como los vectores de la transformación (sería equivalente a una correlación entre imágenes).

En la figura 5.33 aparecen los resultados para subespacios de distintas dimensiones (2, 4, 6 , 10, 20 y 30). Al estar el sistema compuesto por 20 objetos, esto equivale a que el numero de componentes de cada subespacio es 40, 80, 120, 200, 400 y 600, respectivamente. El comportamiento de cada uno de los métodos de extracción de características frente a la dimensión de los subespacios es bastante diferente: para el caso de las imágenes originales, la tasa de error en el reconocimiento se mantiene similar en todos los casos, disminuyendo ligeramente conforme aumenta la dimensión. Sin embargo, PCA cambia su comportamiento según la dimensión utilizada, de manera, que con dimensiones menores (4 y 6) obtienen los errores más pequeños.

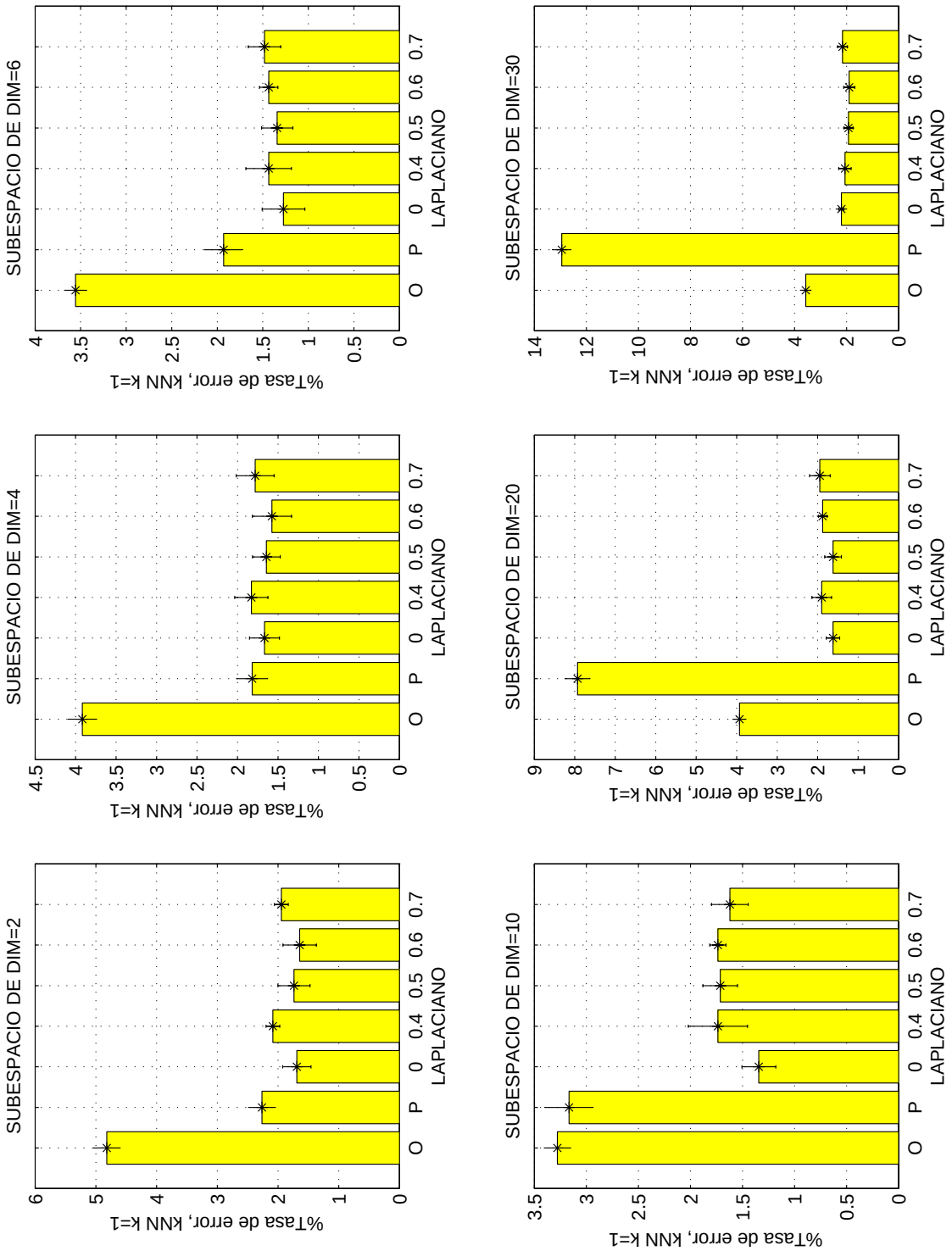
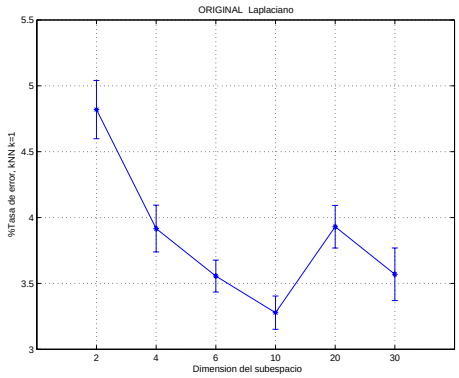
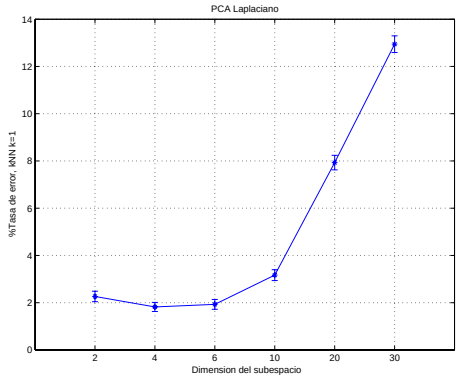


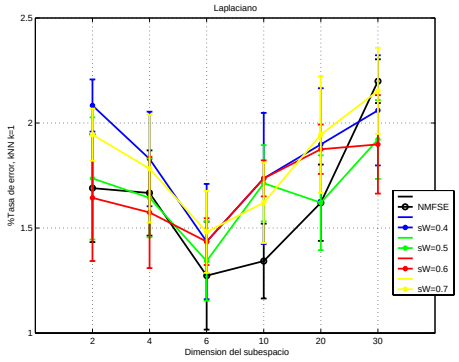
Figura 5.33: Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal L_{ON-OFF}.



(a) $ORIGINAL, \quad L_{ON-OFF}$



(b) $PCA, \quad L_{ON-OFF}$



(c) $NMFSC, \quad L_{ON-OFF}$

Figura 5.34: Comparativa con subespacios objeto modificados en el canal laplaciano (*ON-OFF-L*).

La tasa de error de los algoritmos de NMFSC suele oscilar ligeramente alrededor del 2 % de error en todos los casos, y suele ser la mejor en comparación con el resto de los métodos cuanto mayor es la dimensión.

Similarmente, se han realizado experimentos equivalentes con el canal LoG_{ON-OFF}. La figura 5.35 recopila los resultados más relevantes de esta comparativa para este canal. Los gráficos de la figura 5.36 reproducen estos mismos pero por separado para cada uno de los métodos de extracción de características. Para este canal, el comportamiento de cada uno de los métodos de extracción de características frente a la dimensión de los subespacios es similar al canal L_{ON-OFF}: para el caso de las imágenes originales disminuye ligeramente conforme aumenta la dimensión; PCA, obtienen los errores más pequeños con dimensiones bajas y, la tasa de error de los algoritmos de NMFSC suele oscilar ligeramente alrededor del 2 % de error en todos los casos.

Para probar la robustez del funcionamiento de los métodos de extracción de características se han repetido los experimentos anteriores, pero esta vez, se han añadido oclusiones parciales a las imágenes de test de los objetos. La figura 5.37 muestra todas las vistas de dos de los objetos de prueba, donde se aprecian las oclusiones generadas sobre las imágenes de test (que van cambiando según el experimento de *crossvalidation*).

Las figuras 5.38 y 5.39 recopilan los resultados obtenidos para ambos canales con oclusiones, y la figura 5.40 muestra los resultados organizados por métodos. En esta última figura, se puede apreciar, como se obtienen comportamientos equivalentes a los gráficos anteriores, salvo con la clara ventaja que las tasas de éxito en el reconocimiento de NMFSC en el canal LoG_{ON-OFF} son mucho más bajas que en el resto de los métodos. Es destacable, que en presencia de oclusiones en las imágenes, el método PCA en el canal LoG_{ON-OFF}, produce su mejor tasa de éxito (menor error) otra vez con dimensiones bajas, mientras que NMFSC aporta los mejores resultados para las dimensiones mayores y su error en el reconocimiento disminuye según aumenta el nivel de *sparseness* empleado, lo que le valida como más adecuado que el algoritmo original de NMF para la tarea de reconocimiento de objetos.

El hecho que el canal LoG_{ON-OFF} produzca mejores resultados que el laplaciano, nos conduce a su elección como canal de información a utilizar en la etapa de búsqueda multiescala del método de detección de objetos propuesto en el capítulo 6 de la presente tesis.

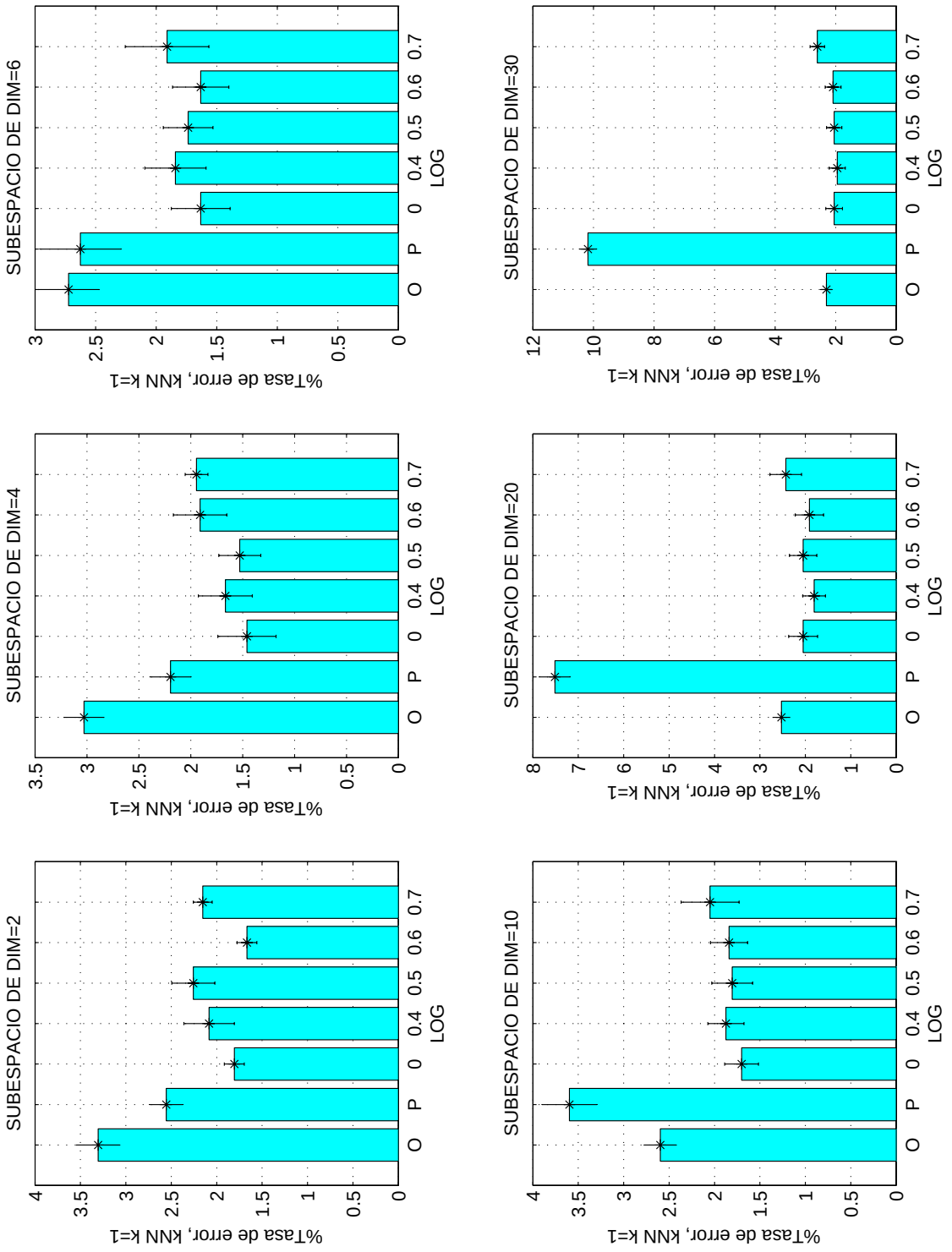
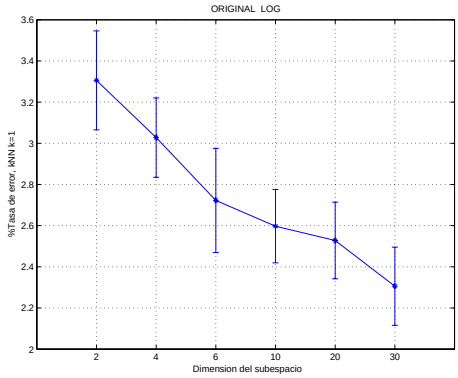
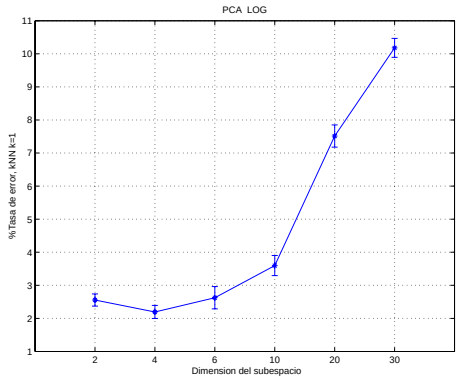


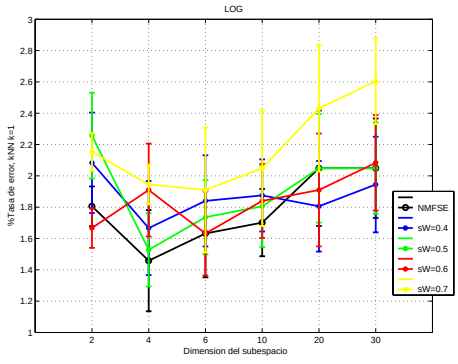
Figura 5.35: Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal LoG_{ON-OFF}.



(a) $ORIGINAL, \quad LoG_{ON-OFF}$

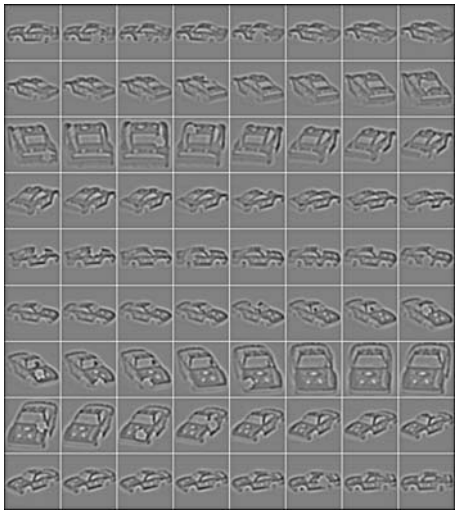


(b) $PCA, \quad LoG_{ON-OFF}$

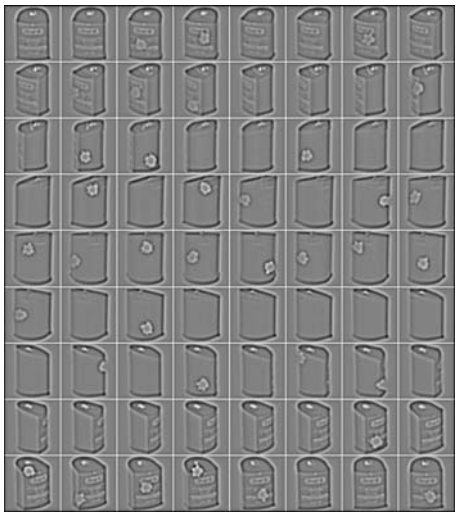


(c) $NMFSC, \quad LoG_{ON-OFF}$

Figura 5.36: Comparativa con subespacios objeto modificados en el canal LoG_{ON-OFF}.



(a) Objeto 15



(b) Objeto 9

Figura 5.37: Imágenes de dos de los objetos de la librería COIL-100 (en el canal L) utilizados durante los experimentos con oclusiones.

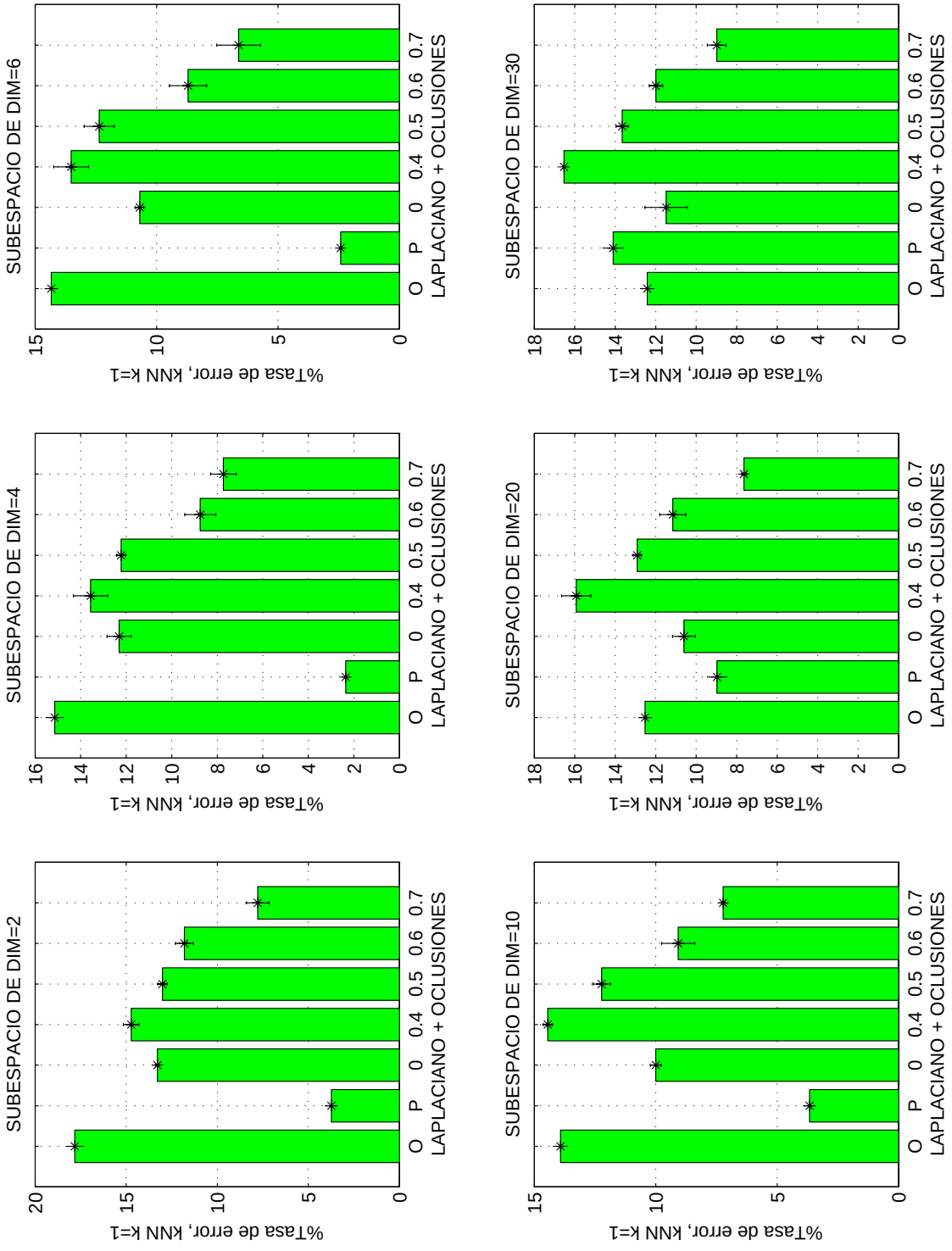


Figura 5.38: Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal L_{ON-OFF}y con presencia de oclusiones en las imágenes de test.

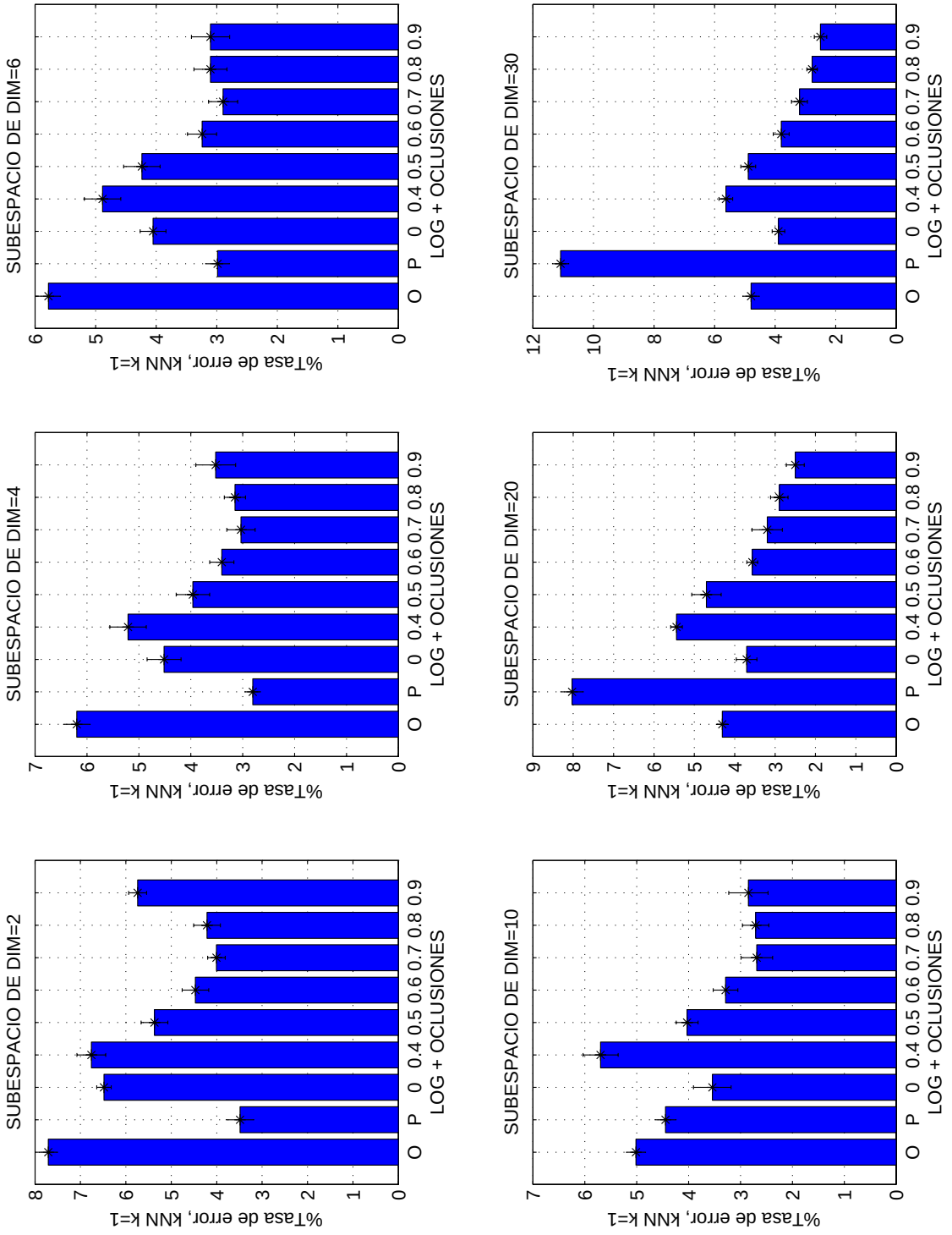
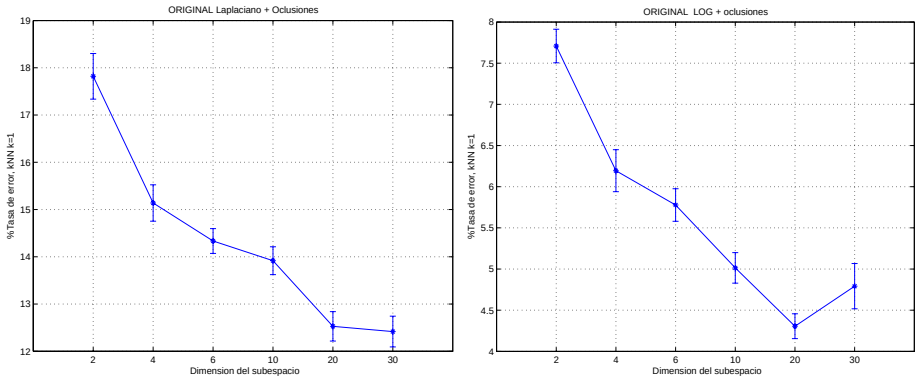
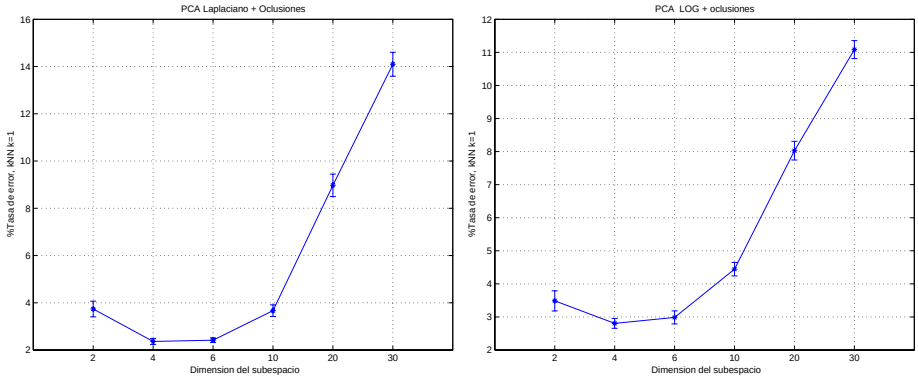


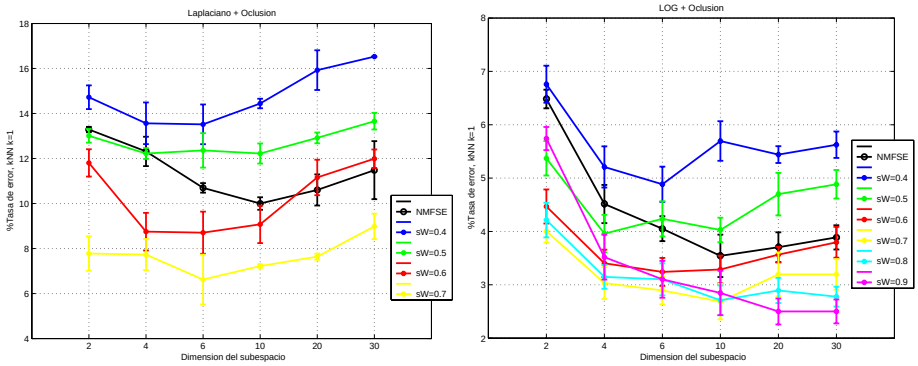
Figura 5.39: Comparativa con subespacios objeto modificados de la técnicas de NMFSC y el resto de métodos en el canal LoG_{ON-OFF} y con presencia de oclusiones en las imágenes de test.



(a) $ORIGINAL, \quad L_{ON-OFF}$ (b) $ORIGINAL, \quad LoG_{ON-OFF}$



(c) $PCA, \quad L_{ON-OFF}$ (d) $PCA, \quad LoG_{ON-OFF}$



(e) $NMFSC, \quad L_{ON-OFF}$ (f) $NMFSC, \quad LoG_{ON-OFF}$

Figura 5.40: Comparativa con subespacios objeto modificados para los dos canales en presencia de oclusiones.

Capítulo 6

Aplicación a la detección de objetos en escenas no estructuradas

Este capítulo describe el sistema de detección de objetos basado en apariencia multicanal y multiescala propuesto, cuya característica principal es su capacidad para trabajar en entornos no estructurados, es decir, sin ningún tipo de control en la geometría o apariencia del objeto, la posición del mismo respecto de la cámara, el fondo de la escena, la iluminación del entorno y la presencia de oclusiones o agrupaciones con otros objetos.

El método de detección propuesto presenta los siguientes aspectos novedosos: realiza la combinación de dos etapas de búsqueda en paralelo; hace uso de dos métodos de extracción de características distintos: PCA y NMFSC; utiliza múltiples canales de información y pondera su peso en la detección final; realiza la combinación de los resultados en función del objeto a detectar; y utiliza escalas o resoluciones distintas para facilitar la búsqueda en entornos complejos donde se desconoce la distancia a la que se encuentra el objeto de interés.

6.1. Tipo de aplicación: detector de objetos basado en apariencia

Los desarrollos teóricos realizados en los capítulos previos se ponen en práctica sobre una aplicación de detección de objetos basada en apariencia. Se ha elegido este tipo de aplicación por ser potencialmente de gran utilidad y también por presentar una dificultad elevada.

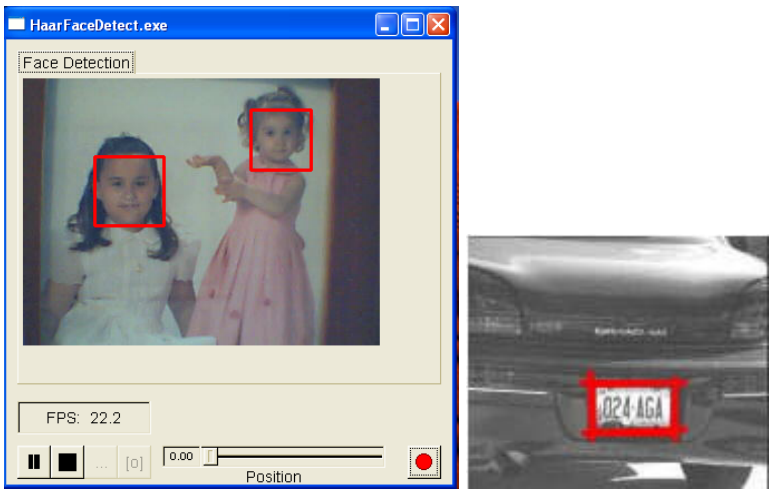
El objetivo final es la detección de un objeto cualquiera en un entorno no estructurado. Es evidente la gran utilidad potencial de una aplicación como la propuesta, dado que permitiría la automatización o robotización de multitud de procesos cuyo principal inconveniente es, actualmente, la incapacidad de funcionar en entornos no estructurados.

En cuanto a la dificultad de la aplicación propuesta, cabe decir que, aunque se trate de tareas que los seres humanos realizamos constantemente y sin aparente dificultad (miramos a nuestro alrededor y detectamos objetos: gente, calles, edificios, automóviles, caras, sillas, etc.), la solución computacional a las mismas presenta grandes dificultades y, de hecho, los desarrollos realizados hasta el presente han sido muy limitados. El principal desafio es la gran variación de la apariencia visual de los objetos. Por ejemplo, para una clase de objetos genéricos[1] como los *automóviles*, éstos varían en forma, color, tamaño, y otros muchos pequeños detalles como los neumáticos o los faros. Además la apariencia visual de un objeto depende de su orientación y de su distancia respecto de la cámara. A esto hay que añadir la ambigüedad de la información visual debida a que las imágenes son proyecciones de dos dimensiones de objetos tridimensionales. Otro factor relevante es el valor de los píxeles, que depende de muchos factores del entorno (intensidad y color de la iluminación, reflectancia del objeto).

Básicamente, la tarea de la *detección de objetos* consiste en localizar regiones en la imagen de test que muestren al objeto de interés. Tradicionalmente, los detectores más populares están orientados a la detección de objetos genéricos como caras [177] [143] y [153] o automóviles y sus matrículas [139]. La figura 6.1 ilustra el funcionamiento de algunos de estos detectores de objetos: un detector de caras [173] muy popular que incorpora la librería OpenCV [83] y un detector de matriculas [133]. Aunque también existen algunos trabajos que estudian el problema intentando obtener métodos válidos para objetos específicos, como en [124] [132] y [145].

Nuestra propuesta consiste en un detector de objetos específicos de forma libre basado en ***apariencia*** utilizando ***características multicanales***, adaptable al objeto, es decir, un algoritmo que permita su uso con objetos distintos. La figura 6.2 ilustra el objetivo que persigue nuestra propuesta.

[1] La definición de objetos *genéricos* y *específicos* se realizó en el apartado 2.1.



(a) Detector de caras de Viola [173] implementado en la librería OpenCV[83].

(b) Detector de placas de matrículas de coches [147].

Figura 6.1: Ejemplos de detectores de objetos genéricos.

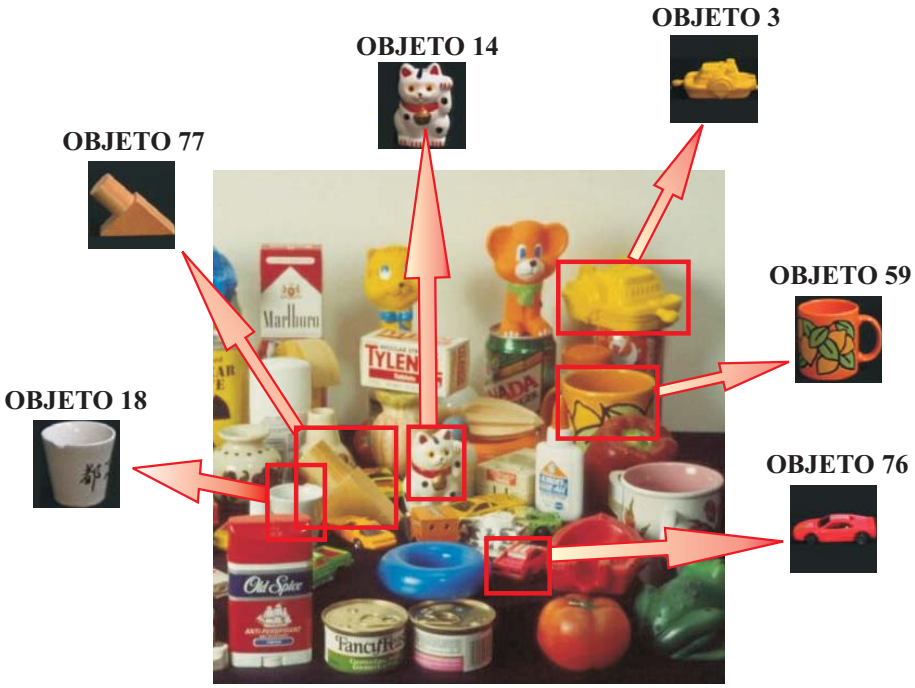


Figura 6.2: Tarea a realizar por el detector de objetos: localizar regiones en una escena que muestren al objeto u objetos de interés. En esta figura se muestra una escena compleja con múltiples objetos de la librería COIL-100.

El algoritmo de detección de objetos basado en apariencia más eficaz, pero de alto coste computacional, consiste, básicamente, en la división de la imagen original en múltiples ventanas de diferentes tamaños (diferentes escalados) para posteriormente, realizar la correlación de cada una de estas ventanas con la imagen o imágenes representativas del objeto [28]. La ventana que presente mayor correlación con la imagen modelo del objeto es la señalada como la que contiene al objeto de interés. La figura 6.3(a) muestra un ejemplo de las ventanas obtenidas al utilizar la figura 6.2 como escena, utilizando un paso de 20 píxeles (en ambos ejes) con un tamaño de ventana de 80×130. En la figura 6.3(b) se muestra un posible modelo de apariencia para el objeto 3 de la librería COIL100: un conjunto de 7 vistas representativas.

Es evidente que este método tan simple funciona eficazmente si el objeto se encuentra aislado del resto de objetos de la escena y el fondo es uniforme y similar al utilizado en las imágenes de entrenamiento (su modelo de apariencia), sin embargo, en escenarios complejos, como el mostrado en la figura 6.2, su funcionamiento puede no ser del todo correcto debido a la presencia de fondos diferentes al de las imágenes de entrenamiento, oclusiones, agrupamientos con otros objetos y diferencias en la iluminación. Otra desventaja considerable es su alto coste computacional: aún estando el objeto aislado en la escena, para su correcta detección se deben obtener ventanas del tamaño adecuado para que el objeto se encuentre presente de forma similar a las imágenes de entrenamiento, por lo que el procedimiento de obtención de ventanas de distintos tamaños hasta conseguir aquella que presente una correlación mayor puede ser largo y tedioso si se utilizan las imágenes de entrenamiento en el espacio imagen, es decir, si no se utiliza ningún tipo de compresión de las ventanas y de las imágenes modelo del objeto.

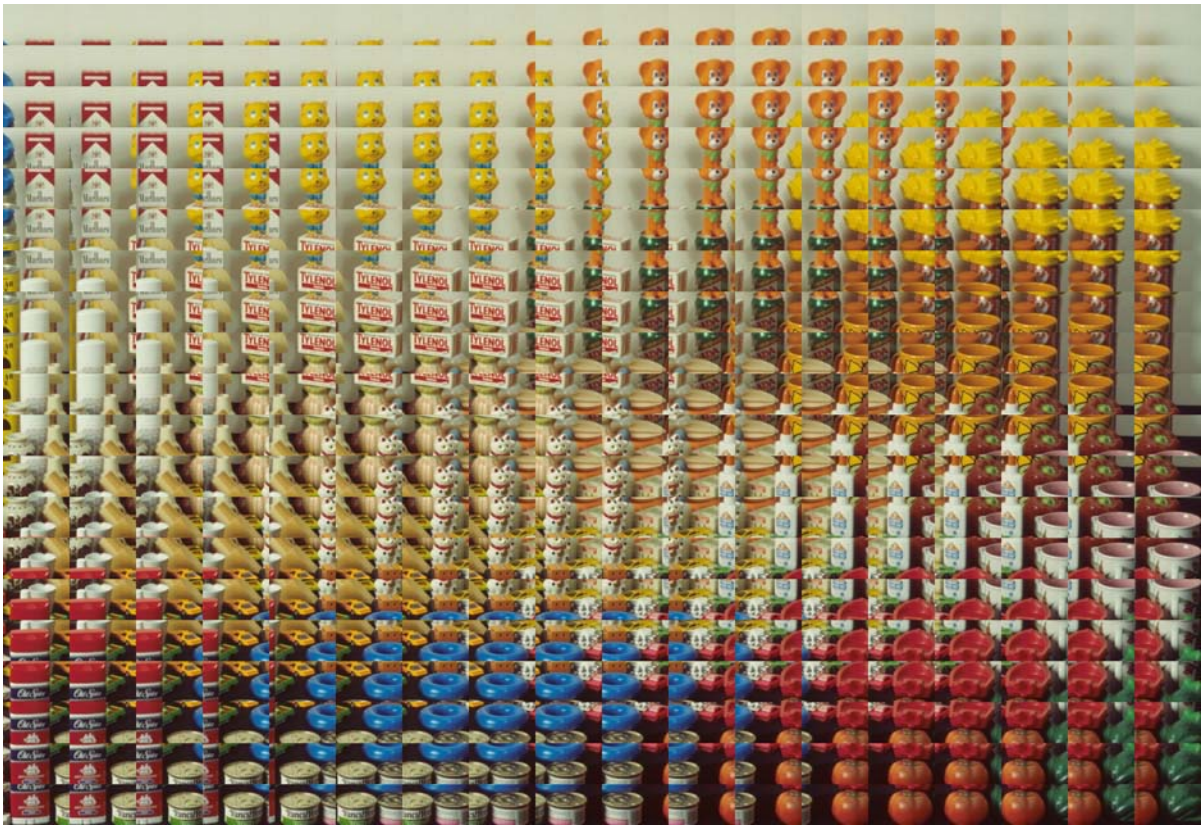
Nuestra propuesta se fundamenta en este método de detección basado en apariencia con ventanas; sin embargo, como se ha visto en capítulos previos, es posible hacer uso de menos información en las imágenes mediante la compresión por extracción de características, lo que reduce drásticamente el coste computacional de los clasificadores utilizados para realizar la correlación, u otra medida de distancia, entre las diferentes imágenes. Las principales novedades del método propuesto son las siguientes:

- Combinación de *dos etapas de búsqueda en paralelo*, para detectar con más robustez y ahorrar coste computacional.
- Utilización de dos métodos de extracción de características distintos: *PCA y NMFSC*.
- Utilización de *múltiples canales de información* y combinación de los resultados en función del objeto a detectar.
- Uso de *escalas o resoluciones distintas* en la escena global para facilitar la búsqueda en entornos complejos donde se desconoce la distancia a la que se encuentra el objeto de interés.

- Adaptación del detector al objeto de interés mediante la ponderación adecuada de los canales de información utilizados.
- Ajuste automático del umbral de detección en función del objeto.

Básicamente, nuestro método de detección visual de objetos está compuesto por la combinación de dos etapas de búsqueda: en una primera etapa, que denominamos *detección multicanal PCA*, se utiliza la información de las imágenes de entrenamiento en varios canales comprimida con PCA, con el objetivo de predecir zonas dentro de la escena donde la probabilidad de que esté el objeto es mayor y, paralelamente, en otra etapa, de *detección multiescala NMFSC*, utilizar la información de las imágenes de entrenamiento comprimidas con NMFSC, a distintas escalas o resoluciones, para localizar el objeto de interés en la escena global. Finalmente, la combinación de ambas etapas permite delimitar la región de la escena en la que se encuentra el objeto.

En los siguientes apartados se explicarán con más detalle cada uno de los pasos del método de detección, que hemos denominado *búsqueda selectiva basada en apariencia multicanal y multiescala*, y se justificará la elección de múltiples canales de información, el uso de los dos métodos de extracción de características antes mencionados y el empleo de múltiples escalas durante la detección del objeto. Algunos trabajos preliminares que recopilan la base del método de detección propuesto se pueden consultar en [162], [163] y [164].



(a) División de la escena en ventanas.

(b) Modelo en apariencia del objeto 3.

Figura 6.3: (a)División de la escena de la figura anterior en ventanas de tamaño 80×130 con un paso de 20 píxeles. La escena original tiene un tamaño de 474×472 píxeles. (b)Un conjunto de 7 vistas representativas del objeto 3 de la librería COIL100.

6.1.1. Descripción de la búsqueda selectiva basada en apariencia multicanal y multiescala

La estructura del algoritmo de detección de objetos propuesto se resume en las figuras 6.4, 6.5 y 6.6, donde las dos etapas antes señaladas se han subdividido para obtener un más alto nivel de detalle del algoritmo. A continuación se explican los contenidos de las mencionadas figuras.

En primer lugar, el gráfico de la figura 6.4 ilustra el proceso de detección de objetos propuesto de manera global. Como ya se ha dicho anteriormente, nuestro método de detección de objetos está compuesto por la combinación de los resultados obtenidos en dos etapas o algoritmos independientes:

Algoritmo de detección multicanal PCA: algoritmo de detección que utiliza como representación visual del objeto la información de las imágenes de varias vistas del objeto, en diversos canales, mediante su compresión con PCA.

Algoritmo de detección multiescala NMFSC: algoritmo de detección que emplea como representación visual del objeto la información de las imágenes de entrenamiento del objeto, en el canal ON-OFF, comprimidas con NMFSC.

Ambos algoritmos disponen de ciertas similitudes y diferencias que se mostrarán a continuación. Dichos algoritmos se han esquematizado en las figuras 6.5 y 6.6 separando los procesos *off-line* y *on-line* que ambos incluyen. Se utilizará está división para una descripción más comprensible de su funcionamiento.

6.1.1.1. Procesos *off-line*

Los procesos a ejecutar *off-line* son aquellos pasos que se efectúan durante el entrenamiento del detector para un objeto de interés en concreto. Dado un cierto objeto, se capturan una serie de imágenes de entrenamiento que muestren todas las posibles vistas diferentes del objeto. En la figura 6.5 aparece un conjunto de 12 imágenes de entrenamiento para el objeto de interés, *la caja amarilla*, que servirá como ejemplo de objeto a detectar a lo largo de este capítulo, para ilustrar el funcionamiento del detector propuesto. A partir de estas imágenes de entrenamiento se construyen dos modelos del objeto: un modelo basado en la apariencia visual, generado utilizando *fragmentos* de las imágenes y reducido con PCA, que se empleará durante la etapa de detección multicanal, y un modelo de apariencia a partir de las imágenes de entrenamiento completas, reducido con NMFSC, que se utilizará en la etapa de detección multiescala.

El conjunto de imágenes de entrenamiento forma el conjunto de información del objeto en el espacio imagen de dimensión igual al número de píxeles de la imagen, esto es, dadas N imágenes de entrenamiento de dimensión M, forman un conjunto de N vectores de dimensión M en el espacio imagen. Como vimos en capítulos anteriores, es posible

reducir la dimensión de estos vectores mediante la técnica de PCA a través de la matriz de transformación $\mathbf{U}^T$ compuesta por los vectores propios de la matriz de covarianzas de los datos centrados en el espacio original. Y ésta es la transformación que se aplica a un conjunto de fragmentos (ventanas o vistas parciales) de las imágenes de entrenamiento[2]. Concretamente, utilizamos una arquitectura tipo subespacio objeto PCA, puesto que todas las imágenes pertenecen a la misma clase, para reducir la información de las imágenes de entrenamiento. Más adelante, en el apartado 6.1.3 se justificará la elección de trabajar con fragmentos de imágenes y no con las imágenes de entrenamiento completas.

La compresión de la información de los fragmentos de las imágenes de entrenamiento se realiza de forma múltiple para cada uno de los distintos canales de información, proceso denominado en la figura 6.5 como *obtención información multicanal*. Esto es, se utilizan las imágenes de entrenamiento filtradas en varios canales: imagen en niveles de gris, imagen en RGB, imagen en H, histogramas en diversos espacios de color, imagen de bordes, etc. Estos canales se describen ampliamente en la sección 6.2.

La figura 6.7 ilustra esta etapa de obtención de los fragmentos (fig.6.7(a)) y el modelo PCA obtenido en un cierto canal de información a partir de la información de los fragmentos de las imágenes (fig. 6.7(b)).

Cada uno de los puntos tridimensionales de la figura 6.7(b) es un fragmento de las imágenes de entrenamiento, en este caso en el canal de apariencia H, reducido a 3 dimensiones mediante la transformación PCA definida por los tres primeros vectores propios de la matriz de covarianzas de los datos. Este conjunto de puntos representa la información visual del objeto en el canal seleccionado, es decir, es el modelo de reconocimiento visual del objeto y se utilizará durante la detección para comprobar si las imágenes de test son semejantes o no a este modelo del objeto en dicho canal. La notación utilizada en la figura 6.5 para designar estos modelos del objeto en cada canal es la siguiente:

$$\{\mathbf{P}_1^*, \mathbf{P}_2^*, \ldots, \mathbf{P}_i^*, \ldots, \mathbf{P}_N^*\} \tag{6.1}$$

siendo i un canal de información dado, N el número total de canales y $\mathbf{P}_i^*$ la matriz en cuyas columnas residen las proyecciones de los fragmentos (puntos tridimensiones en la figura 6.7(b)).

Se generarán tantos modelos PCA como canales de información se utilicen[3], siendo la dimensión de las proyecciones PCA un parámetro ajustable dentro del algoritmo, a mayor dimensión, menor compresión de la información pero menor error de reconstrucción de los datos [90].

[2]Exactamente, la transformación utilizada es la definida en la ecuación 5.6: PCA + Escalado.

[3]En las figuras C.1, C.2, C.3, C.4, C.5, C.6 y , C.7 del anexo C se muestran todos los modelos PCA del objeto *caja amarilla*.

Durante la etapa *off-line* de detección multiescala NMFSC también se genera un modelo del objeto en apariencia, $\mathbf{P}^*$, pero esta vez la transformación utilizada es NMFSC, y sólo en un canal de información, el doble canal *ON-OFF*, que ya apareció en el capítulo anterior y que se detalla más adelante en la sección 6.2.4.

También *off-line* y mediante aprendizaje automático se obtienen una serie de parámetros que se utilizarán durante las etapas *on-line* de ambos algoritmos: los *pesos* de los canales, que ponderan la importancia de cada canal durante la detección del objeto, y el *umbral* de detección, que permite la segmentación del objeto en la escena. Estos dos parámetros se detallarán en las secciones 6.3 y 6.4, respectivamente.

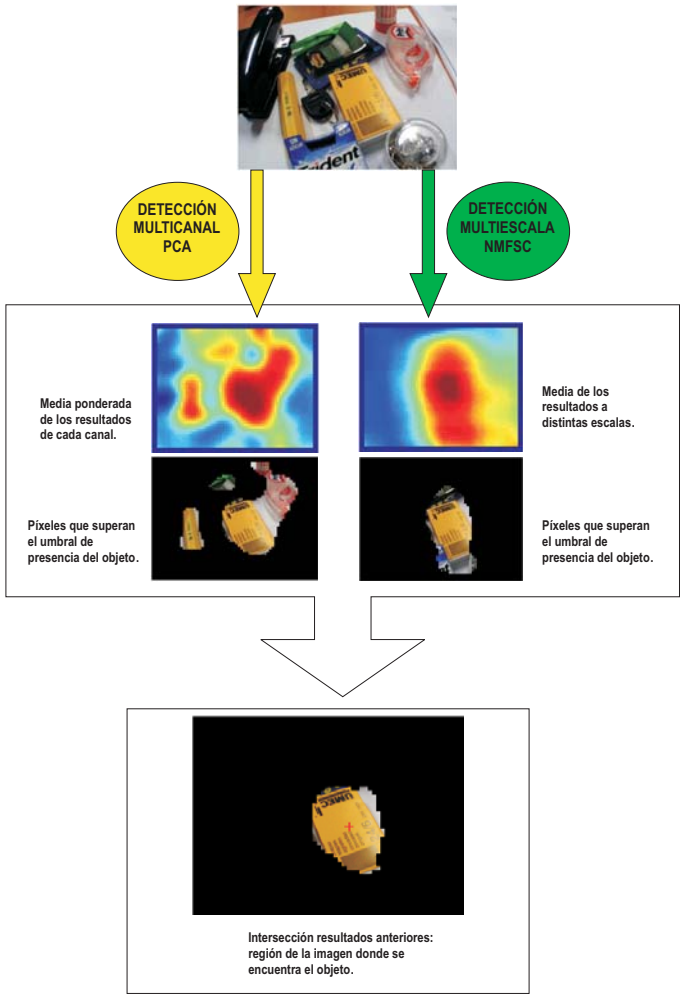


Figura 6.4: Estructura general del algoritmo de detección de objetos propuesto.

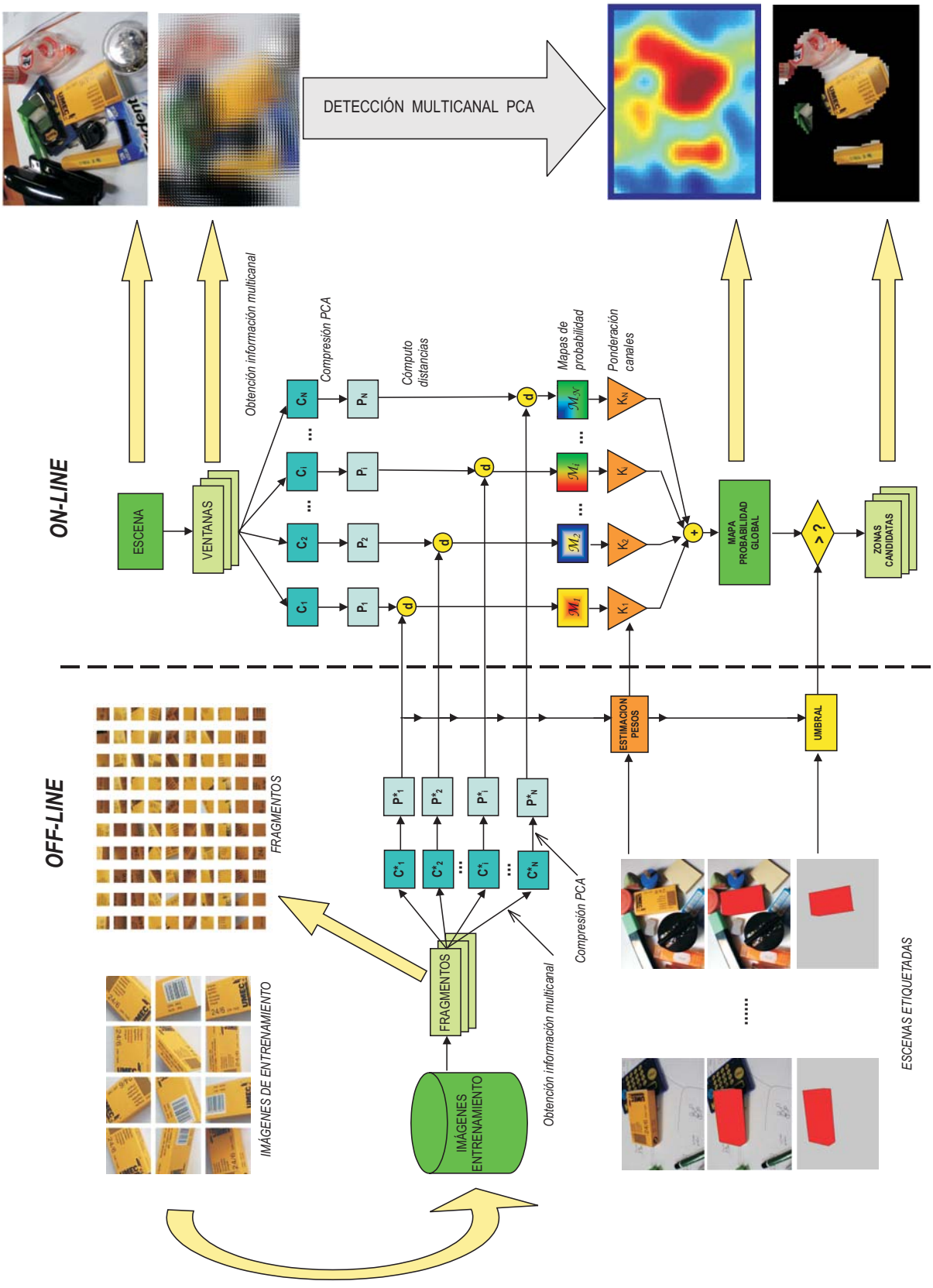


Figura 6.5: Estructura de la detección multicanal PCA.

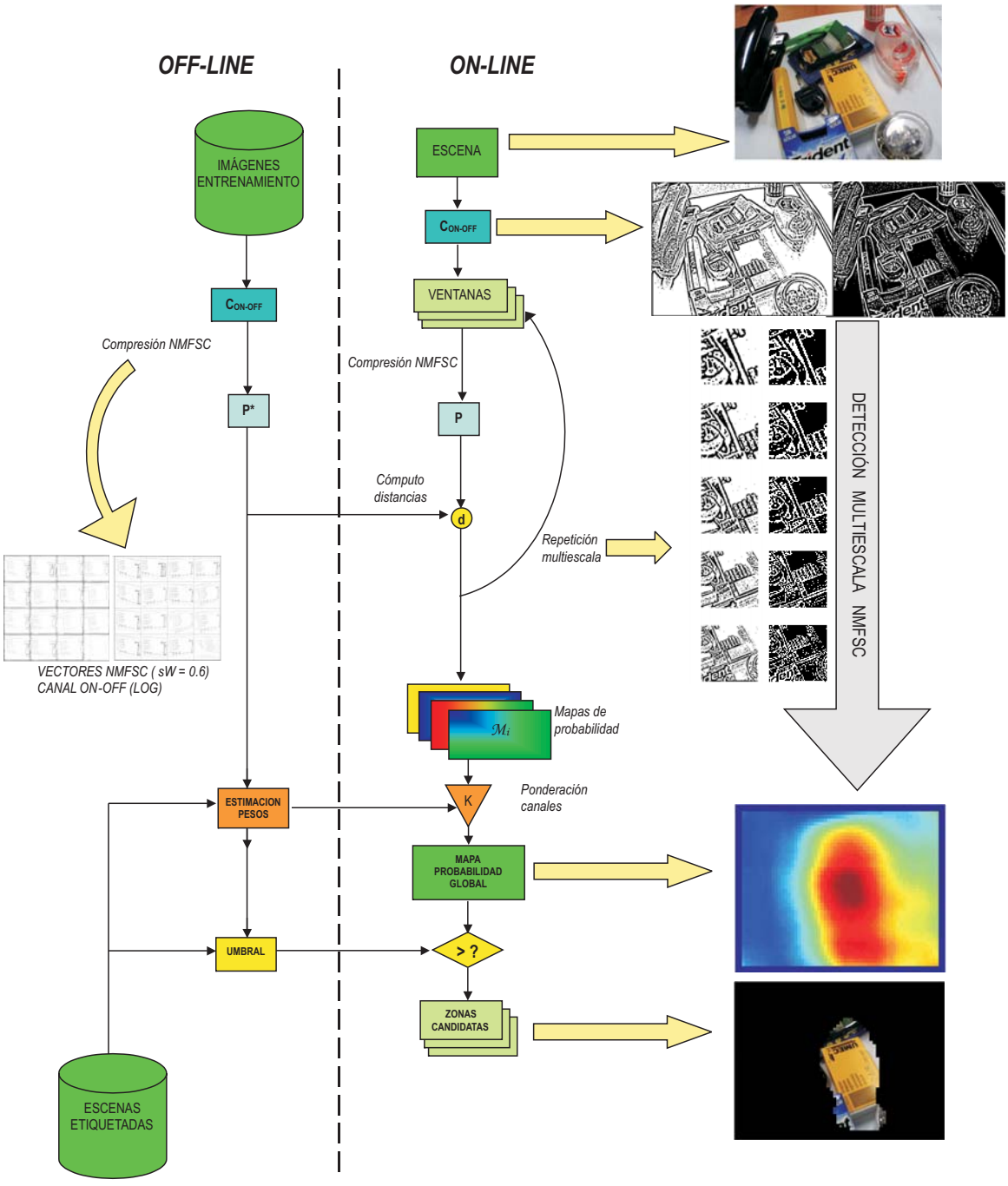
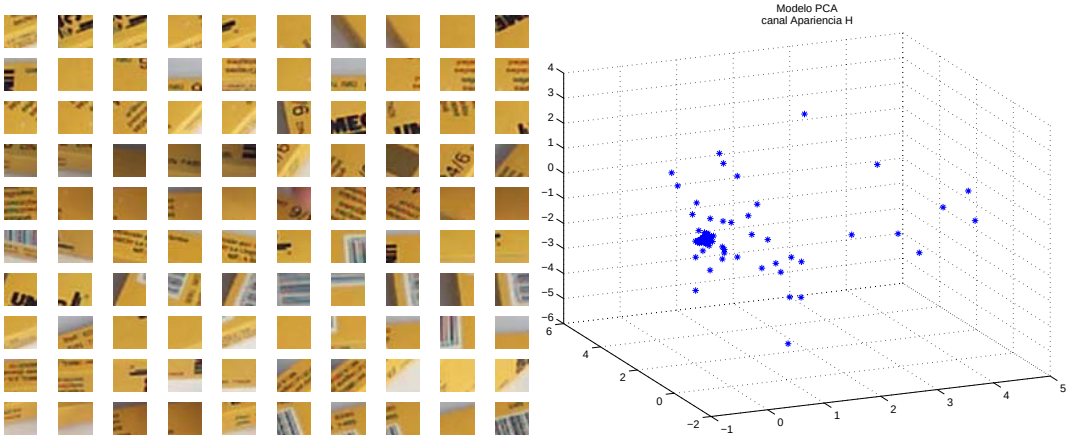


Figura 6.6: Estructura de la detección multiescala NMF.



(a) Fragmentos (b) Modelo 3D PCA

Figura 6.7: Los fragmentos obtenidos a partir de las imágenes de entrenamiento del objeto se comprimen utilizando PCA.

6.1.1.2. Procesos *on-line*

Los procesos *on-line* de ambas etapas, multicanal PCA y multiescala NMFSC, permiten realizar la detección del objeto de interés en modo automático. Cada uno de ellos proporciona un mapa de probabilidad de la escena en función de la posible presencia del objeto de interés. La figura 6.4 muestra los mapas de probabilidad globales obtenidos tras aplicar cada una de estas etapas a la escena que se muestra en la parte superior de la figura. Debajo de cada mapa de probabilidad aparecen visibles las regiones de la escena original que superan un cierto umbral de probabilidad y que acotan la zona donde se encuentra el objeto de interés (*la caja amarilla*) permitiendo su segmentación respecto de la escena. En la parte inferior de la figura se muestra la combinación (*intersección*) de dichas etapas: se ha conseguido delimitar con mayor precisión la posición del objeto en la escena global.

La figura 6.5 ilustra la fase *on-line* de la detección multicanal PCA: la escena se divide en múltiples ventanas, $\mathbf{v}_k$, por cada uno de los canales de información utilizados (*obtención de información multicanal*, tal como se denomina en la figura 6.5), y estas ventanas son reducidas dimensionalmente mediante las matrices de transformación obtenidas durante el entrenamiento para cada canal i (*compresión PCA*). Siendo $\mathbf{P}_i$ la matriz de las proyecciones de las ventanas de la escena en el canal i. Las proyecciones de las ventanas, $\mathbf{P}_i$, se comparan con los modelos obtenidos *off-line*, $\mathbf{P}_i^*$, utilizando una medida de distancia (*cómputo distancias*); en concreto, se ha utilizado la distancia Euclídea, que junto con la transformación PCA empleada equivale a la distancia de Mahalanobis, puesto que el

método de PCA utilizado incluye el escalado posterior con los valores propios, esto es, se utiliza el proceso de *whitening* descrito en la ecuación 5.6.

De este modo, por canales, a cada ventana $\mathbf{v}_k$ se le asocia un valor igual a su distancia mínima al modelo, que será inversamente proporcional a la probabilidad de que el objeto de interés se encuentre en la ventana considerada:

$$(\mathbf{v}_k)_i \Leftarrow \text{mín}(dist[(\mathbf{p}_k)_i, \mathbf{P}_i^*]) \tag{6.2}$$

donde el índice i indica el canal de información, k es el índice de la ventana y $\mathbf{p}_k$ es la proyección de la ventana centrada, $\hat{\mathbf{v}}_k$, en el subespacio PCA correspondiente al canal i:

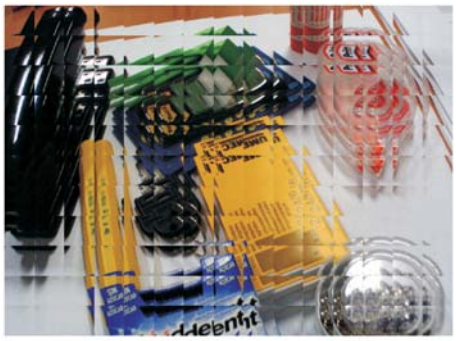
$$(\mathbf{p}_k)_i = \widetilde{\mathbf{\Lambda}}^{-1}(\widetilde{\mathbf{U}}^T)_i \cdot (\hat{\mathbf{v}}_k)_i \tag{6.3}$$

En la figura 6.5 se emplea $\mathbf{P}_i$ para denotar la matriz que reúne todas las proyecciones de las ventanas, $(\mathbf{p}_k)_i$, en el canal i considerado.

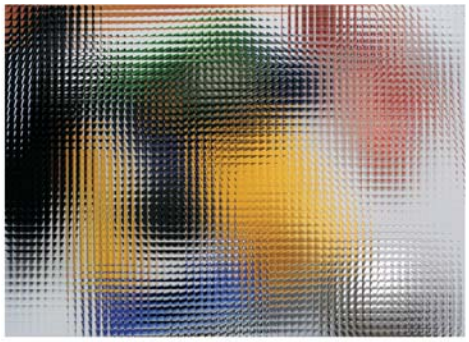
Según el número de veces que un mismo píxel aparece en varias ventanas, es posible asignarle un valor de distancia al modelo, realizando el promedio entre todas las distancias asociadas a él en cada una de las ventanas en la que es utilizado. Esta asignación entre píxeles de la escena y su distancia respecto al modelo del objeto depende del paso y el tamaño de la ventana utilizado, puesto que la fragmentación de la escena en múltiples ventanas y por tanto el número total de ventanas, N_v,depende de estos dos parámetros.

Las figuras 6.8 y 6.9 ilustran este concepto de la división de la escena en múltiples ventanas según el paso utilizado en el barrido y el tamaño de las ventanas. En la figura 6.8 aparecen tres barridos de la escena distintos según pasos y tamaños de ventanas diferentes; y en la figura 6.9 aparecen las matrices de repetición de píxeles asociadas a cada uno de los barridos, y que indican las veces que un mismo píxel aparece en varias ventanas. Un paso muy pequeño en relación con el tamaño de la ventana, como es el caso de las figuras 6.8(b) y 6.9(b), da lugar a que un mismo píxel se utilice en muchas ocasiones, unas 50 veces para la mayoría de los píxeles en el caso de la figura 6.9(b). Y el fenómeno opuesto, que un mismo píxel sea visitado poco frecuentemente, ocurre al utilizar un paso considerablemente grande en comparación con el tamaño de la ventana (figuras 6.8(c) y 6.9(c)).

Es importante señalar que la combinación de estos dos parámetros (paso y tamaño de las ventanas) influye drásticamente en el coste computacional de las etapas de detección como se mostrará en el apartado 6.1.4.



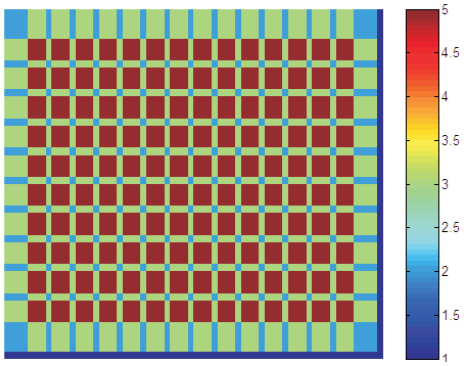
(a) Tamaño ventana = 70×70, paso = 40



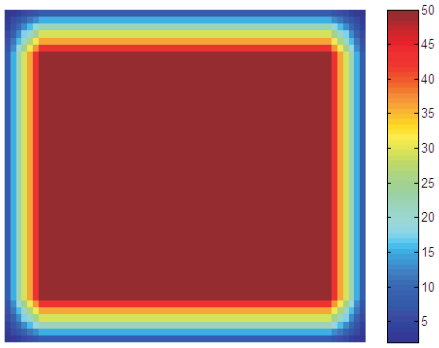
(b) Tamaño ventana = 70×70, paso = 10

(c) Tamaño ventana = 40×40, paso = 30

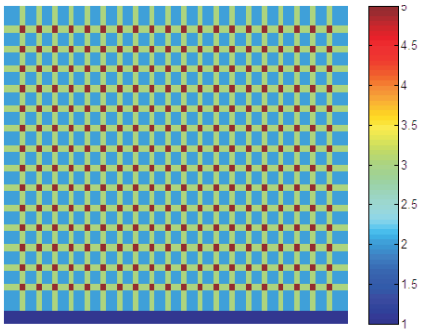
Figura 6.8: La división de la escena en ventanas depende del tamaño de la ventana y el paso utilizado para recorrer toda la imagen. El tamaño y paso de la ventana está expresados en número de píxeles.



(a) Tamaño ventana = 70×70, paso = 40



(b) Tamaño ventana = 70×70, paso = 10



(c) Tamaño ventana = 40×40, paso = 30

Figura 6.9: Las matrices de repetición de píxeles indican las veces que un mismo píxel aparece en varias ventanas y cabe tenerlas en cuenta durante la construcción de los mapas de probabilidad de la escena.

Continuando con la descripción de la etapa de detección multicanal PCA, mediante la asignación de distancias a cada píxel de la imagen, se consigue obtener una imagen en distancias inversamente proporcional a la probabilidad que el píxel pertenezca al objeto de interés. Esta imagen, matriz o mapa de distancias se denotará como $\mathcal{D}_i$. Con el fin de poder combinar los mapas de distancias obtenidos en cada uno de los distintos canales, es necesario realizar la normalización de dichos mapas de distancias. Este proceso de normalización puede realizarse según la función representada en la figura 6.10: una distancia al modelo nula indica probabilidad máxima (1) ante el hecho que el píxel pertenezca al objeto, mientras que se asignará probabilidad nula a la distancia máxima obtenida, bien en la escena o bien en las distancias respecto del valor central en el modelo de apariencia (ver apartado 6.4). Para distancias intermedias, el valor de probabilidad se obtiene mediante una relación lineal como la mostrada en la figura. Al mapa de distancias normalizado se le denominará a partir de ahora mapa de probabilidad, $\mathcal{M}_i$.

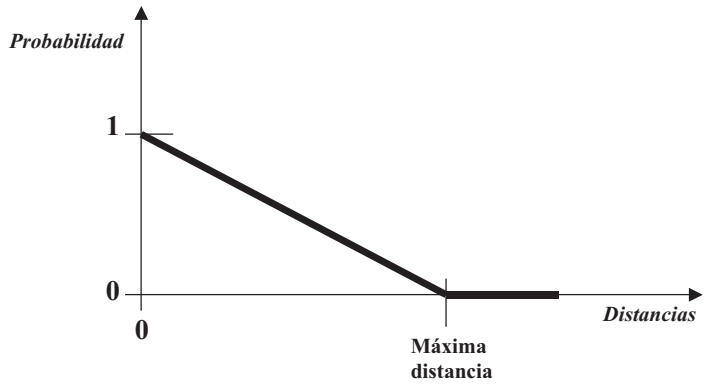


Figura 6.10: Normalización de los mapas de distancia.

Tal como se aprecia en la figura 6.5, se obtienen tantos mapas de probabilidad del objeto en la escena, como canales de información se consideren. Posteriormente, se realiza una ponderación de estos mapas, para obtener un mapa de probabilidades global de la escena, $\mathcal{M}$, siendo la siguiente expresión la relación entre canales más sencilla a considerar:

$$\mathcal{M} = \frac{K_1 \cdot \mathcal{M}_1 + K_2 \cdot \mathcal{M}_2 + \ldots + K_N \cdot \mathcal{M}_N}{K_1 + K_2 + \ldots + K_N} = \frac{\sum_{i=1}^{N} K_i \cdot \mathcal{M}_i}{\sum_{i=1}^{N} K_i} \tag{6.4}$$

En el apartado 6.3 se discutirá la forma de obtener automáticamente los pesos K_i asociados a cada uno de los canales.

En la parte inferior derecha de figura 6.5 se muestra un ejemplo de mapa de probabilidades global. Las áreas en rojo, señalan las zonas de mayor probabilidad en las que puede encontrarse el objeto; mientras que las áreas en azul indican la ausencia del objeto en esas

zonas de la escena. Este mapa es el resultado final de la etapa de detección multicanal PCA y acota las regiones en la escena donde puede hallarse el objeto.

De forma equivalente a la etapa de detección multicanal PCA, en la etapa *on-line* de detección multiescala NMFSC se utiliza el modelo del objeto $\mathbf{P}^*_{NMFSC}$ para localizar la zona dentro de la escena en la que el objeto se encuentra con mayor probabilidad. En esta etapa sólo se utiliza un canal de información visual, el canal ON-OFF, pero en cambio, el barrido de la escena se repite varias veces con tamaños de ventanas distintos con el objetivo de obtener diversos mapas de probabilidad en función del tamaño de la ventana utilizado (o la operación equivalente: utilizar una ventana de tamaño fija, pero emplear escenas escaladas a distintos tamaños). Este hecho permitirá la localización del objeto aunque su posición respecto de la cámara no sea la misma que la utilizada durante la captura de las imágenes de entrenamiento. Este proceso de repetición de la búsqueda con tamaños de ventanas distintos se ha denominado en la figura 6.6 *repetición multiescala*. En dicha figura se muestra a modo de ejemplo un conjunto de 5 ventanas, en el canal ON-OFF, a distintas escalas, de forma que en cada par de ellas se aprecia como la escala de búsqueda del objeto en la escena es diferente.

Paralelamente a la etapa previa, los mapas de probabilidad obtenidos en cada una de las diferentes escalas de búsqueda se combinan finalmente para obtener un mapa de probabilidad global de la escena mediante la expresión 6.4. En el proceso de combinación de resultados de la búsqueda en múltiples escalas, el valor de los pesos de los canales a distintas escalas depende de la aplicación. Por ejemplo, si se desconoce la distancia del objeto a la cámara, todos los canales se ponderan de igual forma, como podría ser el caso en una aplicación de navegación robótica, en la que el robot en principio debe localizarse dentro de una zona determinada. Sin embargo, si la distancia del objeto a la cámara se conoce, como sería el caso, por ejemplo, de la detección de objetos que circulan sobre una cinta transportadora delante de un robot fijo en una posición de la planta, se daría un peso mayor a las escalas que permiten detectar mejor el objeto, pudiendo incluso eliminar de la búsqueda algunas escalas muy lejanas a la posición esperada del objeto. En el apartado 6.3 se justificará la selección de los pesos de los canales para ambas etapas del algoritmo de detección.

Este mapa de probabilidad global es el resultado final de la etapa de detección multiescala NMFSC y junto con el mapa de probabilidad global de la etapa de detección multicanal PCA servirá para determinar con mayor robustez y exactitud la posición del objeto y segmentarlo adecuadamente respecto del resto de la escena.

6.1.1.3. Combinación de las etapas multiescala y multicanal

Cada una de las etapas de detección proporcionan un mapa de probabilidad relativo a la escena que informa sobre la mayor o menor probabilidad de que un determinado píxel pertenezca o no al objeto de interés. Además, cada mapa de probabilidad lleva asociado un parámetro umbral que permite decidir si el objeto está o no presente en la escena, y en caso positivo, determinar en qué zona en concreto. En la sección 6.4 se presentarán dos metodologías que permiten la obtención automática de dicho umbral.

Luego, asumiendo que cada mapa de probabilidad lleva asociado un umbral que permite decidir sobre la localización del objeto en la escena, cada una de las etapas producirá una escena segmentada en la que se visualizará el área correspondiente al objeto, tal como se muestra en las dos imágenes umbralizadas de la figura 6.4.

El aumento o disminución de cada uno de los umbrales en sus respectivos mapas de probabilidad produce que el área dentro de la escena asociada al objeto sea mayor o menor, con la consiguiente inclusión de más falsos positivos o bien el aumento de falsos negativos.

Nuestra propuesta consiste en la combinación de las escenas umbralizadas de ambas etapas de forma que disminuyan los falsos positivos de cada una de las dos etapas por separado. Esta combinación consiste, básicamente, en realizar la intersección entre los píxeles de ambas escenas segmentadas o umbralizadas, con el objetivo de obtener mayor certeza de la presencia del objeto en los píxeles resultantes, puesto que serán píxeles positivos de pertenencia al objeto, proporcionados por dos métodos de compresión muy diferentes. Esta forma de combinar los resultados de ambas etapas requiere que los umbrales utilizados sean lo suficientemente permisivos, de forma que se eviten en la medida de lo posible los falsos negativos. La figura 6.4 muestra en su parte inferior el resultado de realizar esta intersección de píxeles entre las dos escenas umbralizadas en cada una de las etapas. Se puede apreciar como se ha reducido claramente el número de falsos positivos. A partir de esta escena final umbralizada se podrá obtener el centro del objeto como el centro de masas de la zona segmentada.

6.1.2. ¿Por qué utilizar varios canales de información?

El sistema de visión humano puede reconocer, sin previo aprendizaje, diferentes vistas de objetos comunes a una velocidad cercana a los 10 objetos por segundo, lo que implica que el reconocimiento de un sólo objeto se realiza en unos 100 ms [137] desde la estimulación de la retina hasta las partes más elevadas del sistema visual. Sin embargo, los estudios neurofisiológicos revelan que la latencia en las respuestas de las neuronas en las áreas del cerebro dedicadas al reconocimiento de objetos son también del orden de los 100 ms [67]. La realidad es que el sistema de visión humano es capaz de reconocer fácilmente una gran variedad de estímulos que corresponden a objetos tridimensionales rígidos, no rígidos, articulados, caras humanas, texturas de superfícies, escenas naturales complejas,.. siendo insensible a cambios en los puntos de vista de los objetos, oclusiones parciales y agrupamientos de muchos objetos en una misma escena. Si el sistema de visión humano funciona tan bien y tan rápido, cuando las unidades de proceso, esto es, las neuronas, tienen tiempos de respuesta lentos, implica una solución de complejidad temporal baja (pocos pasos) pero una alta complejidad espacial, es decir, muchos procesos en paralelo, que está respaldada en términos biológicos con el enorme número de conexiones sinápticas que tiene cada neurona. Es razonable suponer que la utilización de múltiples canales de información imita en cierto modo el paralelismo realizado por el sistema de visión humano y por lo tanto, puede ofrecer buenos resultados.

En correspondencia con los axiomas del reconocimiento de patrones estadístico, el reconocimiento visual de objetos en el cerebro es el proceso de proyectar o enlazar los píxeles de la retina en un espacio de características más conveniente (que los píxeles) que permita la clasificación invariante de los objetos según diferentes vistas. Dentro de este espacio de características, representado por la actividad de las neuronas, la similitud entre la vista de un objeto y otro viene dada por un sencillo cálculo de distancias. El reconocimiento de una entrada se consigue encontrando la clase más similar almacenada en memoria, que corresponde a una vista del objeto previamente aprendida. El espacio de características debe cumplir que grandes cambios en la orientación del objeto (diferentes vistas) produzcan pequeños desplazamientos en el espacio, mientras que, pequeños cambios en la calidad del objeto (forma, textura, color) produzcan grandes desplazamientos en el espacio de características, dando lugar al reconocimiento de otros objetos.

Además, cabe considerar, que el sistema de visión humano es capaz de reconocer objetos o escenas a partir de cualquier tipo de vista, aún no habiendo sido aprendida con anterioridad. Esto implica, que no es necesario conocer todas las vistas de un objeto para su aprendizaje y posterior reconocimiento [118].

Evaluando la plausibilidad de este esquema de extracción de características como modelo de la visión biológica humana, Bartlett [13] propuso las siguientes propiedades que deben tener las características construidas por el sistema de visión humano: deben ser *numerosas* para propiciar el sistema de procesado en paralelo, *útiles* (sensibles a la calidad

del objeto, pero no a su pose o configuración) y obtenidas a través de *diferentes canales* de información, como color, forma, textura,..

De manera similar, en el método propuesto incluimos el uso de diferentes canales de información con el fin de mejorar la capacidad de reconocimiento del sistema, aunque ello suponga un incremento del coste computacional. Este incremento no sería tal en el caso de una implementación paralela del método propuesto en un sistema multiprocesador. La facilidad con la que se puede realizar tal implementación es una de las ventajas del método.

Al utilizar más información del objeto, se consigue aumentar la robustez del sistema de detección, pues la combinación de todos los canales puede solventar el mal funcionamiento de algunos de ellos, bien porque se producen demasiados falsos positivos o todo lo opuesto, porque la zona donde reside el objeto es descartada en algún canal. Además, el uso de varios canales permite adaptar el sistema de detección al objeto. Esta adaptación del algoritmo al objeto se realiza a través de los coeficientes K_i y se discute en profundidad más adelante en el apartado 6.3.

6.1.3. ¿Por qué utilizar fragmentos de las imágenes de entrenamiento?

El motivo principal de utilizar fragmentos de las imágenes de entrenamiento y no las imágenes completas para la generación de los modelos PCA del objeto en la etapa de detección multicanal, es que los fragmentos o características de complejidad intermedia[4] son óptimos para el reconocimiento visual de objetos. Ullman *et al.* [158] demuestran que los fragmentos de imágenes de tamaño intermedio son más informativos para la clasificación visual de objetos que los fragmentos muy simples (puntos o líneas de cierta orientación en los objetos [127]) o muy complejos (imagen global del objeto [118][157]).

Esta afirmación de la superioridad de los fragmentos para clasificación es justificada mediante dos tipos de análisis computacionales sobre un conjunto de unos 10000 fragmentos pertenecientes a dos tipos de objetos (clases): 138 *caras* frontales y 40 vistas laterales de *automóviles* (junto con un conjunto de imágenes no perteneciente a ninguna clase, la clase rechazo, formado por imágenes de paisajes, rutas y juguetes). Los fragmentos son obtenidos de las distintas imágenes, en diferentes posiciones, tamaños y resoluciones.

El primero de los análisis consiste en el cálculo de la cantidad de información en relación a su clase residente en cada fragmento. Esta medida se realiza utilizando la ecuación de información mutua:

$$I(C,F) = H(C) - H(C|F) \tag{6.5}$$

[4] Intermediate complexity (IC) features [158].

donde $I(C, F)$ es la información mutua entre el fragmento F y la clase C de las imágenes, siendo H la entropía. En [158], la cantidad de información mutua es denominada *merit*. Las figuras 6.11(b) y 6.11(d) muestran el mejor conjunto de 8 fragmentos obtenidos (con mayor *merit*) para la clase de las caras. La figura 6.11(c) muestra un conjunto de fragmentos seleccionados para la clase de los coches. Si se dibuja un gráfico del valor de *merit* en función del tamaño del fragmento se obtiene una curva similar a los gráficos con el símbolo $\triangle$ de las figuras 6.11(e) y 6.11(f): donde el valor máximo se encuentra en los fragmentos de tamaño intermedio.

El segundo análisis que se propone en [158] para confirmar la superioridad de los fragmentos en clasificación es una medida que indica la utilidad del fragmento para clasificación. Esta medida, denominada *weight*, se expresa como:

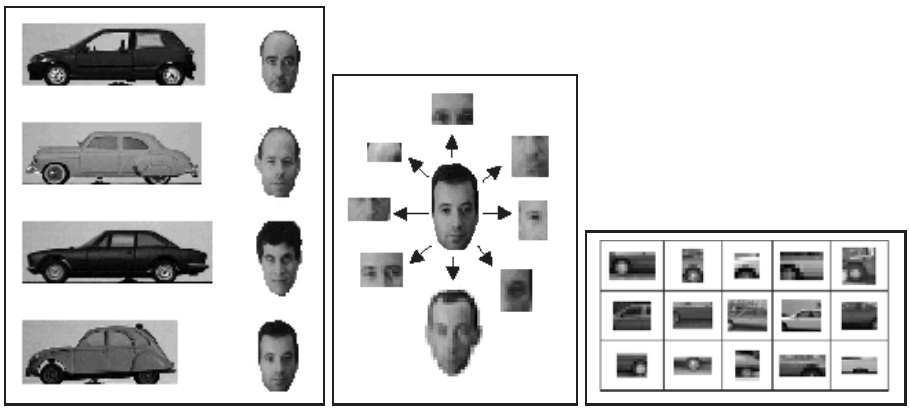
$$w = log_2(R(F)) \tag{6.6}$$

donde $R(F)$ es un cociente muy utilizado en procesado de señales, y aquí se utiliza como criterio de detección óptimo para la presencia de un objeto de la clase C dado el fragmento F:

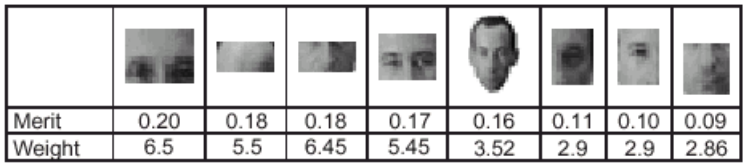
$$R(F) = \frac{P(F|C)}{P(F|\overline{C})} \tag{6.7}$$

El cálculo de las medidas de *weight* de los fragmentos proporciona los mismos resultados que la información mutua según se aprecia en las figuras 6.11(e) y 6.11(f) con las curvas con símbolo $\square$: se obtienen los valores máximos para los fragmentos de tamaño intermedio.

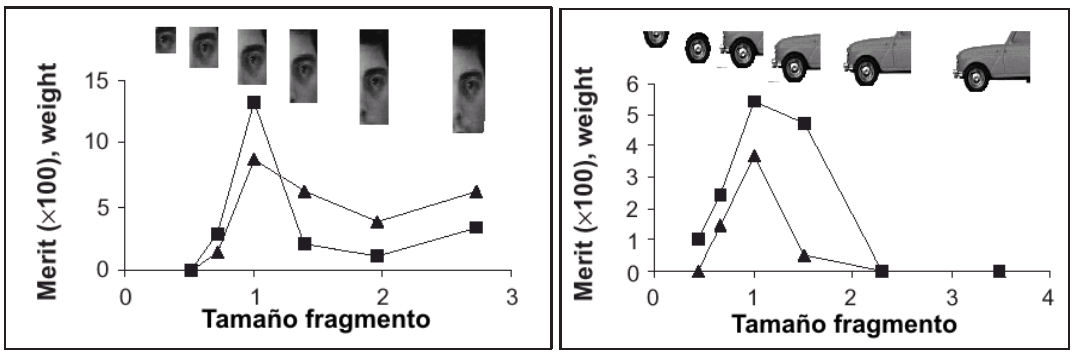
La superioridad de los fragmentos de tamaño intermedio en el reconocimiento de objetos genéricos, como los coches y las caras, puede explicarse debido a la interacción de dos factores: especialización y frecuencia relativa. Un fragmento grande de una clase (digamos una cara) puede proporcionar indicios fiables de la presencia de una cara en una imagen, sin embargo, la probabilidad de encontrar dicho fragmento en una cara distinta es pequeña. Consecuentemente, la información de ese fragmento respecto a la clase está limitada. Un fragmento más pequeño, tiene una probabilidad mayor de aparecer en imágenes de caras distintas, pero la probabilidad de que también aparezca en imágenes de no-caras es también muy alta.



(a) (b) (c)



(d)



(e) (f)

Figura 6.11: Resumen de los principales resultados del trabajo de Ullman *et al.* sobre el contenido de información en los fragmentos de imágenes de caras y coches. (a) Ejemplos de las imágenes de caras y coches utilizadas para generar los fragmentos. (b) Los mejores 8 fragmentos alrededor de una cara, en orden decreciente de información mutua desde arriba y en sentido contrario a las agujas del reloj. (c) Un conjunto de fragmentos de imágenes de coches. (d) Los 8 mejores fragmentos de caras con su valor de *merit*(△) y *weight*(□). Superioridad de las características de complejidad intermedia: *Merit* y *weight* de los fragmentos de caras (e) y coches (f) en función del tamaño del fragmento.

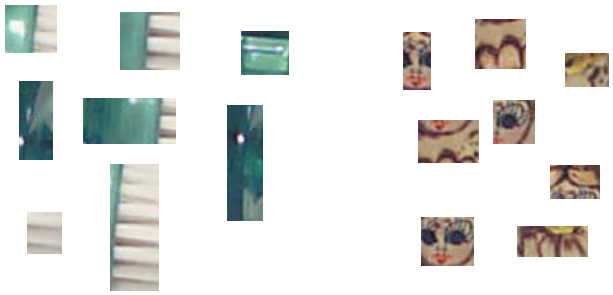
Además, el uso de fragmentos permite realizar la búsqueda del objeto en la escena con ventanas de forma cuadrada (o rectangular), aunque el objeto disponga de una apariencia bidimensional muy irregular. Este hecho facilita la programación del algoritmo de búsqueda, puesto que si se utilizara la imagen completa del objeto para construir el modelo habría que emplear ventanas con forma irregular y dependiente del objeto o bien utilizar ventanas rectangulares y asumir que la ventana de búsqueda considerará zonas que no pertenecen al objeto debido a su forma no rectangular. Este fenómeno se ilustra en la figura 6.12, donde se muestran las imágenes completas (incluido su fondo) de dos objetos cuya proyección en dos dimensiones no es una forma rectangular. La figura 6.12(b) muestra el área que ocupa el objeto en la imagen y cuya geometría podría utilizarse para buscar el objeto en una escena compleja, no obstante, esta forma es intrínsecamente dependiente del objeto y de la posición del objeto respecto de la cámara. Sin embargo, la figura 6.12(c) muestra un conjunto de fragmentos rectangulares que podrían emplearse para la construcción del modelo de apariencia del objeto y permitirían el uso de ventanas rectangulares durante la búsqueda del objeto en la escena.

En esta tesis, el análisis de Ullman *et al.* [158], acerca de la superioridad de los fragmentos frente a las imágenes completas de los objetos, se ha enfocado bajo el marco del reconocimiento de objetos específicos. De esta forma, se ha utilizado la función de *merit* para analizar la importancia del tamaño de los fragmentos de las imágenes de los objetos específicos en su detección. Sin embargo, por simplicidad, se ha restringido el proceso de fragmentación de las imágenes de entrenamiento y sólo se han utilizado fragmentos cuadrados.

La figura 6.13 muestra un gráfico con el valor de merit obtenido en función de la longitud en píxeles del fragmento para el objeto *caja amarilla* en el canal de información I. Para obtener esta curva se han generado 200 fragmentos pertenecientes a la clase del objeto (a partir de 20 imágenes de entrenamiento correspondientes a vistas distintas del objeto de interés), y 800 fragmentos no pertenecientes a la clase (a partir de imágenes de otros objetos distintos); y con esta información se ha utilizado el algoritmo descrito en [158] para el cálculo del *merit* sobre un conjunto de 30 imágenes de test. En la figura 6.13 es posible apreciar que para los fragmentos de menor tamaño los valores obtenidos son muy pequeños y, a partir de un tamaño de fragmento de unos 30 píxeles, el valor de información mutua empieza a aumentar considerablemente. El valor máximo se obtiene para una longitud de 50 píxeles, y a partir de ese valor comienza a decrecer. En la parte inferior del gráfico se muestran algunos fragmentos de la clase *caja amarilla* utilizados durante el análisis.

(a) Imágenes completas.

(b) Área que ocupa el objeto.



(c) Fragmentos.

Figura 6.12: El uso de fragmentos facilita la programación del algoritmo de búsqueda: (a) imágenes completas de dos objetos; (b) el área que ocupan en la imagen; (c) fragmentos cuadrados y rectangulares de los objetos, que permiten el uso de ventanas regulares durante la búsqueda del objeto en la escena.

Este mismo estudio se ha repetido con otros nueve objetos aparte del objeto *caja amarilla*. La figura 6.14 muestra la media de *merit* y su desviación típica en función de la longitud en píxeles del fragmento para estos diez objetos. De esta forma, el valor medio de *merit* de los fragmentos de estos 10 objetos refleja con más claridad la importancia de los fragmentos de tamaño intermedio.

El buen funcionamiento de los fragmentos de tamaño intermedio para representar los objetos en la tarea de detección de objetos se mostrará experimentalmente en el siguiente capítulo. En la implementación de nuestro algoritmo de detección de búsqueda selectiva basada en apariencia multicanal, el tamaño de los fragmentos es un parámetro ajustable durante la etapa de entrenamiento, y los resultados obtenidos confirman la superioridad de utilizar fragmentos de tamaño intermedio en vez de las imágenes completas de los objetos, facilitando la detección del objeto en imágenes donde su pose es diferente a la de las imágenes de entrenamiento. También se mostrará como el uso de fragmentos facilita la detección del objeto en escenarios muy complejos, con agrupaciones de muchos objetos y oclusiones parciales del objeto de interés.

Sin embargo, durante la fase de entrenamiento de la etapa de detección multiescala NMFSC no se utilizan fragmentos de imágenes, sino las imágenes completas de los objetos. Esto es así por la naturaleza de la transformación NMFSC. Como se vió en el capítulo anterior, los vectores proporcionados por el método NMFSC proporcionan características locales de los objetos. Además los resultados de reconocimiento muestran la idoneidad de la técnica NMFSC cuando las imágenes de test presentan oclusiones parciales del objeto de interés. Si se utilizaran fragmentos de las imágenes para calcular esta transformación, no se obtendrían las características locales que son las responsables del buen funcionamiento de este método de compresión ante oclusiones parciales de los objetos.

Por tanto, aunque el entrenamiento de la etapa de detección multiescala NMFSC se realiza con las imágenes completas de los objetos, los vectores de características utilizados son de tipo local, como en el caso de utilizar los fragmentos de las imágenes con la técnica de PCA. Ambos métodos, intentan construir los modelos de apariencia de los objetos a partir de información local o parcial.

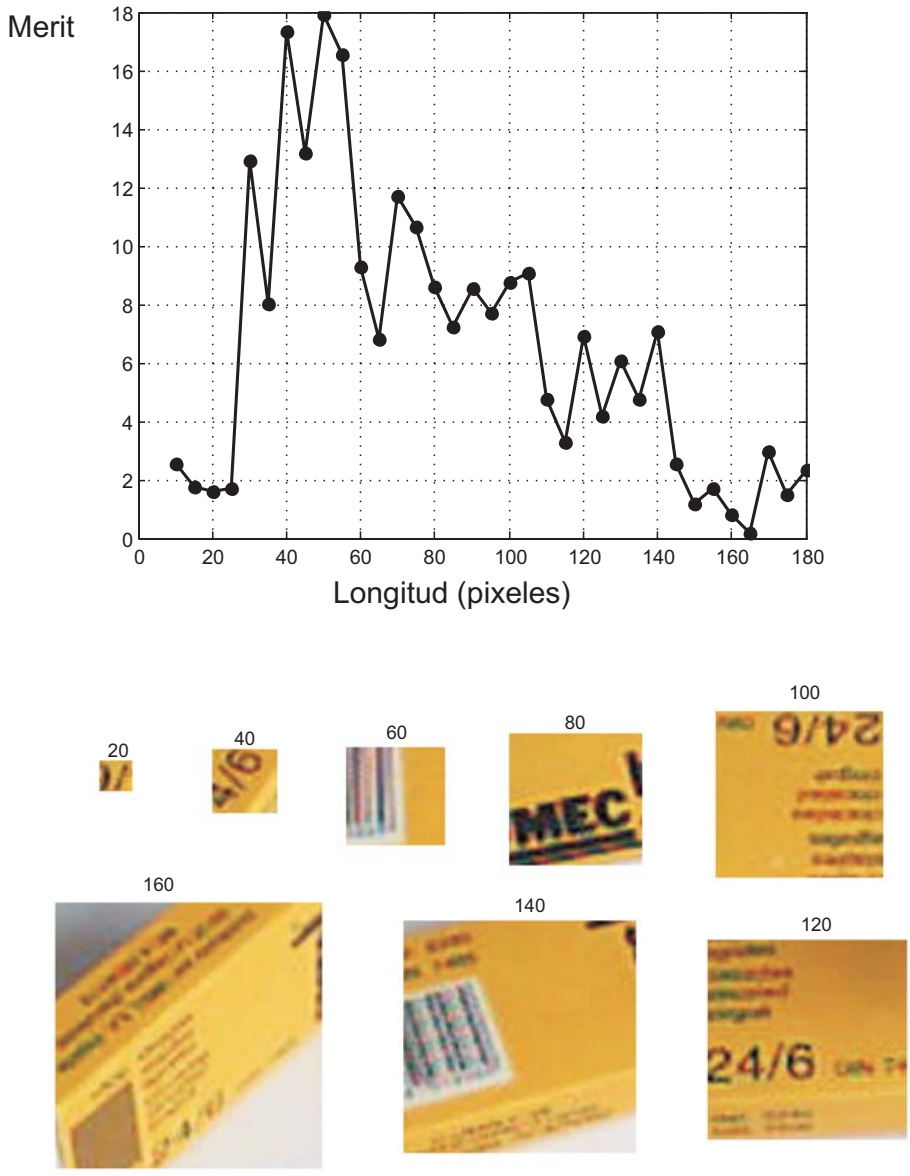


Figura 6.13: *Merit* de los fragmentos de la caja amarilla en función del tamaño del fragmento.

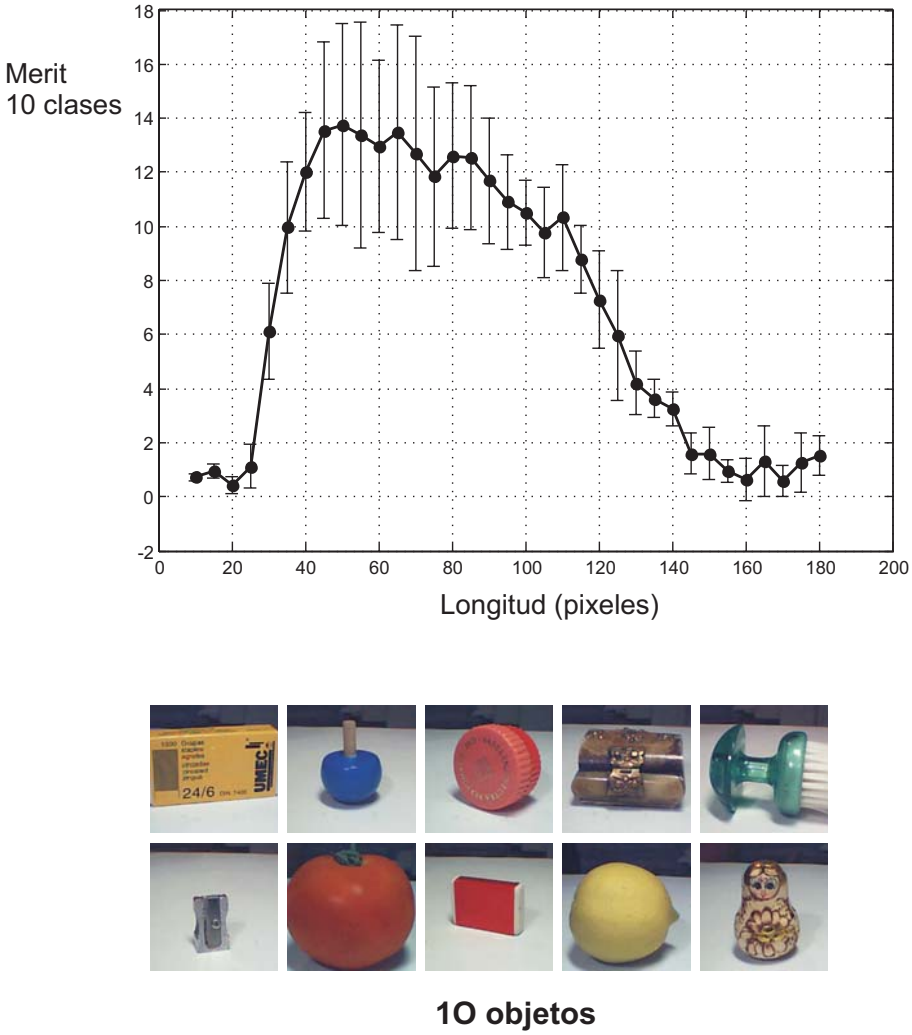


Figura 6.14: *Merit* promediado de los fragmentos de 10 objetos en función del tamaño del fragmento.

6.1.4. ¿Por qué utilizar dos tipos de compresión?

El hecho de utilizar dos técnicas de extracción de características diferentes para la compresión de la información visual del objeto: los fragmentos de las imágenes de entrenamiento con PCA durante la detección multicanal y las imágenes completas con NMFSC durante la detección multiescala, se fundamenta en los siguientes aspectos:

- Por un lado, evidentemente, nuestro propósito es utilizar múltiples canales de información y por otro, utilizar múltiples escalas. Un procedimiento de fuerza bruta consistiría en evaluar todos los canales sobre todas las escalas, pero el coste computacional de tal procedimiento sería excesivo.

  El número total de ventanas, N_v, en el que se divide una escena durante la etapa de búsqueda del objeto, depende del paso y del tamaño de la ventana utilizado de la siguiente forma:

$$N_v = \left\lfloor \frac{N_x - w_x}{p_x} + 1 \right\rfloor \cdot \left\lfloor \frac{N_y - w_y}{p_y} + 1 \right\rfloor \tag{6.8}$$

  donde N_x y N_y es el número de píxeles de la imagen, w_x y w_y es el tamaño de la ventana rectangular en píxeles y, p_x y p_y es el paso del barrido en ambos ejes. La figura 6.15 ilustra este proceso de división de la escena en múltiples ventanas. A partir de la expresión anterior, se puede apreciar cómo el número total de ventanas crece de forma inversamente proporcional al paso utilizado. El gráfico de la figura 6.16 ilustra este efecto, donde se ha tomado $p_x = p_y = [10 : 1 : 100]$, $w_x = w_y = 60$, $N_x = 480$ y $N_y = 640$. Por lo tanto, el número de ventanas se incrementa muy rápidamente con la disminución del tamaño del paso utilizado, lo que produce que aumente considerablemente el coste computacional de la búsqueda, pues cada ventana deberá proyectarse en el subespacio de cada canal y luego compararse (mediante el clasificador) con las proyecciones modelo.

  El número total de ventanas también depende del tamaño de las mismas según refleja la expresión 6.8. La figura 6.17 muestra esta relación con los siguientes valores: $w_x = w_y = [10 : 1 : 100]$, $p_x = p_y = 10$, $N_x = 480$ y $N_y = 640$. Ahora vemos un comportamiento similar, aunque lineal, el número total de ventanas crece gradualmente según disminuye el tamaño de las ventanas. Por lo que utilizar ventanas muy pequeñas aumenta el coste computacional de la búsqueda.

  Esta dependencia del coste computacional del método de búsqueda sobre la escena con el número de ventanas utilizadas, se hace aún más patente si se realiza una búsqueda multiescala de la escena, esto es, el barrido de la escena se repite varias veces con tamaños de ventanas distintos, o lo que es equivalente, para un tamaño de ventana fijo, la imagen es escalada (Este efecto se ilustra en la figura 6.18.).

  Por otra parte, de entre los métodos de compresión evaluados, PCA resulta muy apropiado en términos de coste computacional y de eficiacia para la mayor parte

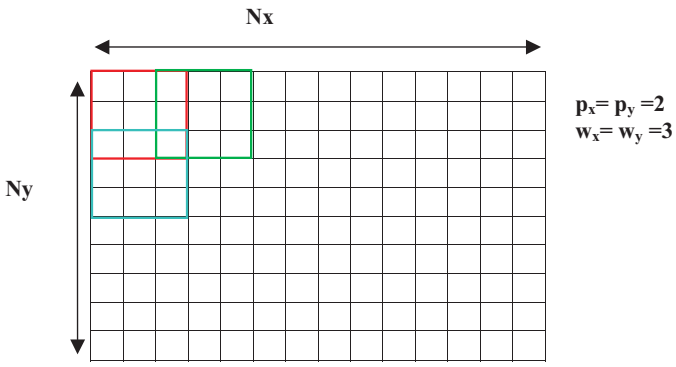


Figura 6.15: División de una escena en múltiples ventanas. Cada cuadrado de la ilustración equivale a un píxel.

de los canales; mientras que NMFSC resulta superior en el caso de existencia de oclusiones. Sin embargo, NMFSC, sólo obtiene resultados óptimos sobre el canal ON-OFF, con lo que no es aplicable a todos los canales. Por tanto, se ha optado por utilizar PCA junto con el algoritmo de detección que hace uso de múltiples canales de información, y emplear NMFSC sólo con el canal ON-OFF pero reutilizando el modelo de apariencia obtenido para repetir la búsqueda del objeto en diferentes escalas o resoluciones de la escena. Con ello, conseguimos ambos propósitos: el uso de información multicanal y la búsqueda del objeto a distintas escalas, sin un coste computacional demasiado elevado en ambas etapas, puesto que la etapa PCA conlleva un coste computacional asociado al número de canales, mientras que la etapa NMFSC asocia su coste computacional con la búsqueda multiescala, de hecho, la relación es proporcional, utilizar los 38 canales en la etapa PCA es aproximadamente equivalente computacionalmente a utilizar una treintena de escalas en la etapa NMFSC.

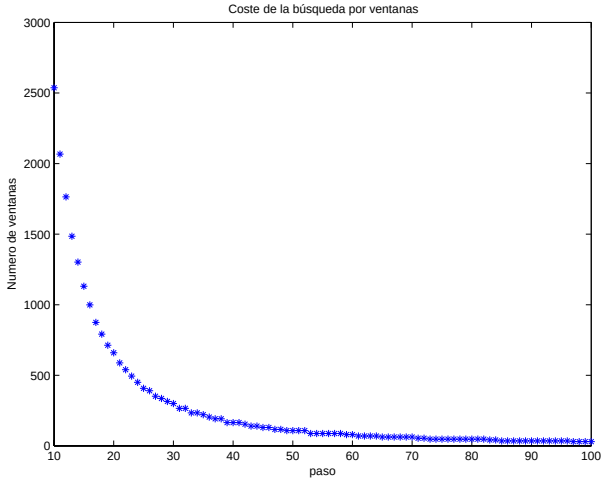


Figura 6.16: Relación entre el número de ventanas y el paso del barrido.

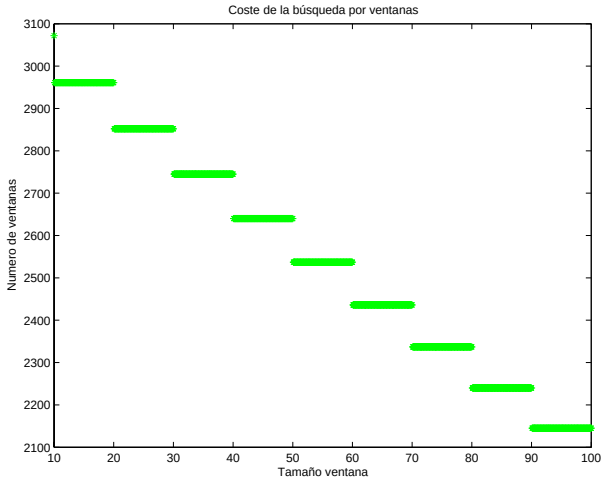


Figura 6.17: Relación entre el número de ventanas y su tamaño.

Figura 6.18: La búsqueda multiescala consiste en el barrido de la escena varias veces con tamaños de ventanas distintos, o de modo equivalente, para un tamaño de ventana fijo, aplicar factores de escalado sobre la imagen original.

- A primera vista parece posible afirmar que la etapa de búsqueda multicanal PCA podría no funcionar correctamente si la distancia del objeto a la cámara es distinta en las escenas de test y en las imágenes generadoras de los modelos. Sin embargo, el hecho de utilizar una amplia gama de fragmentos, ayuda a identificar el objeto dentro de un rango de distancias moderado, puesto que el hecho de utilizar fragmentos, información local, permite localizar el objeto sin necesidad de estar a una distancia completamente fija respecto de la cámara. Ambas etapas, hacen uso de características locales, bien a través de los fragmentos o bien a través de la información aportada por la transformación NMFSC, permitiendo cierta flexibilidad en las condiciones ambientales durante la búsqueda del objeto. Claro está que si el objeto se encuentra a una distancia muy lejana a las imágenes utilizadas durante el entrenamiento, la etapa multicanal dejará de funcionar correctamente si no se hace uso de la búsqueda multiescala, aunque esto, evidentemente lleva asociado un aumento del coste computacional.

- Otro aspecto importante a considerar es que al utilizar dos métodos de extracción de características totalmente diferentes y combinar sus resultados, se consigue aumentar la robustez del sistema de detección y eliminar posibles falsos positivos. Al utilizar la información visual del objeto de interés comprimida mediante alguno de los métodos de extracción de características y una medida de distancia como clasificador se obtendrán numerosos falsos positivos, pues el contenido de una ventana puede ser totalmente diferente a cualquiera de las vistas del objeto y sin embargo su proyección en el subespacio objeto puede estar muy cerca de los puntos que representan al objeto y por tanto, su valor de probabilidad ser alto. El hecho de emplear dos modelos del objeto generados con técnicas diferentes, y en varios canales de información en el caso de la detección multicanal PCA, disminuye la proporción

de falsos positivos al combinar el resultado de los diferentes mapas de probabilidad, pues unos mapas descartarán los falsos positivos de otros. Este problema de la obtención de falsos positivos con los subespacios objeto apareció en el apartado 4.2, las figuras 4.7, 4.8 y 4.9 ilustran este concepto. De modo casual, es posible que la proyección de las imágenes de dos objetos diferentes coincidan. En el siguiente capítulo se mostrarán los resultados experimentales obtenidos al estudiar la capacidad de detección de la combinación de ambas etapas.

6.2. Canales de información utilizados en la búsqueda selectiva multicanal y multiescala

A continuación se describen cada uno de los canales de información utilizados durante las dos etapas de detección. Durante la etapa de detección multiescala nuestro sistema incluye un total de 38 canales, que se pueden agrupar en tres tipos: canales de apariencia[5], canales de histogramas y canales de forma o bordes. En la etapa multiescala sólo se utiliza un canal, el canal ON-OFF, puesto que es el idóneo al procesar la información con NMFSC.

6.2.1. Canales de apariencia

Los canales de apariencia están compuestos por los fragmentos de las imágenes de entrenamiento de cada objeto en varios espacios de color, reordenando los píxeles de la imagen en un vector columna de dimensión igual al número de píxeles del fragmento.

Existe una gran diversidad de espacios de color: RGB, CMY (para visualización e impresión); YIQ, YUV (para television y video); XYZ (conjunto estandar de colores primarios; rgb, xyz (colores normalizados); U*V*W*, L*a*b*, Luv (espacios perceptuales uniformes); HSI y HSV (para los humanos). Muchos de ellos son combinaciones lineales de RGB, como CMY y XYZ; o están normalizados respecto de I en analogía al rgb, como IQ, xyz, UV, U*V*, a*b*, uv. Por lo tanto, para nuestro sistema sólo se utilizarán aquellos canales de color esencialmente diferentes. De esta forma, los espacios de color típicos que puede emplear nuestro algoritmo durante las etapas *off-line* y *on-line* de la detección multicanal PCA son el modelo RGB normalizado, que denominaremos ***rgb***, el modelo ***HSV*** y el canal de intensidad ***I***. Además se han incluido dos nuevos espacios de color, $\mathbf{l_1 l_2 l_3}$ y $\mathbf{c_1 c_2 c_3}$, desarrollados por Gevers y Smeulders [66] y que presentan mayor robustez para el reconocimiento de objetos. A continuación, se definen los espacios de color propuestos para nuestro detector.

6.2.1.1. RGB y rgb

El espacio de color más utilizado en sistemas de visión para representar información en color es el espacio RGB (*Red-Green-Blue*). Este espacio está basado en la síntesis aditiva de la Teoría tricromática del color desarrollada por Young. (Cualquier radiación del espectro visible (color) puede generarse por la combinación de 3 finas franjas espectrales

[5]Todos los canales pueden denominarse en general, de *apariencia*, puesto que utilizan la información de la imagen del objeto como un vector multidimensional en un espacio de dimensiones igual al número de píxeles de la imagen; sin embargo, en la clasificación de canales, se ha preferido etiquetar como canales de apariencia sólo a aquellos en los que no se utiliza otra transformación intermedia entre la imagen en un determinado espacio de color y el método de extracción de características. No obstante, cabe recordar que todos ellos proporcionarán un modelo de reconocimiento basado en *apariencia* del objeto en cuestión.

monocromáticas: Rojo (700nm), Verde (520 nm) y Azul (400nm), ver figura 6.19). El espacio de color RGB se utiliza normalmente en los monitores de tubos de rayos catódicos (CRT), televisiones, scanners y cámaras digitales. Para un monitor la luminiscencia de un fósforo consiste en la suma de colores primarios: simplemente los colores se parametrizan a través de los coeficientes α, β y γ, de manera que cada color se forma como:

$$C = \alpha R + \beta G + \gamma B \tag{6.9}$$

El valor de los coeficientes varía desde 0 (no luminiscencia) a 1 (valor máximo del fósforo)[6]. Con esta parametrización las coordenadas de color forman un volumen cúbico donde se sitúan en los vértices el color negro [0 0 0], los tres colores primarios (rojo [1 0 0], verde [0 1 0] y azul [0 0 1]), las tres mezclas secundarias (cyan [0 1 1], magenta [1 0 1] y amarillo [1 1 0]) y el blanco [1 1 1], tal como se muestra en la figura 6.20(a).

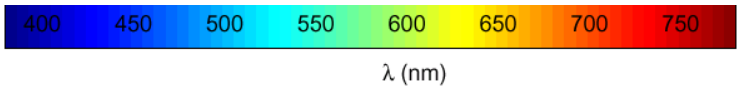
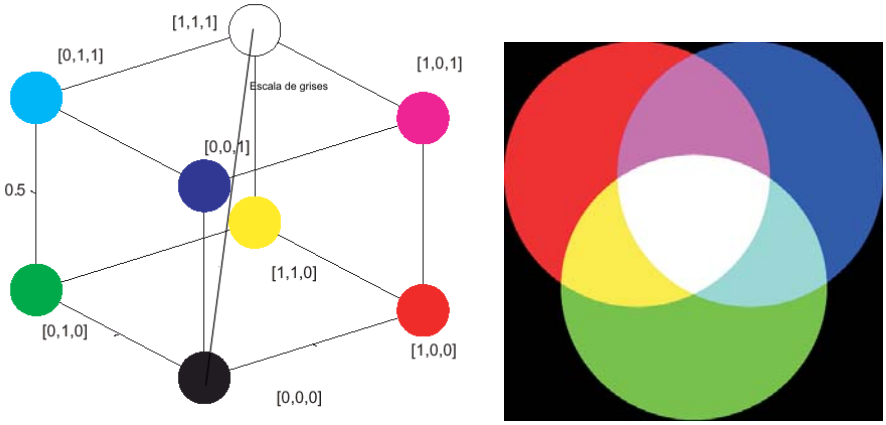


Figura 6.19: Espectro de luz visible.



(a) RGB (b) Mezcla aditiva de colores

Figura 6.20: Modelo de color RGB.

La diagonal principal que conecta los vértices negro y blanco define el canal de intensidad:

[6] 1 o 255, según la escala utilizada.

$$I(R,G,B) = R + G + B \tag{6.10}$$

Todos los puntos en un plano perpendicular al eje de los grises del cubo de color RGB tienen la misma intensidad. Este plano atraviesa al cubo de color en los puntos $R = G = B = M$ y recorta un triángulo equilatero que define al modelo rgb:

$$\begin{aligned} r(R,G,B) &= \frac{R}{R+G+B} \\ g(R,G,B) &= \frac{G}{R+G+B} \\ b(R,G,B) &= \frac{B}{R+G+B} \end{aligned} \tag{6.11}$$

Utilizando el modelo de reflexión de materiales diélectricos no homogéneos bajo iluminación blanca [146], se deduce que el espacio de color RGB tiene una gran varianza a lo largo de un superficie curva de color uniforme, es decir, una imagen de un objeto de color uniforme no presenta uniformidad en el espacio RGB. El mismo argumento se mantiene para I. Sin embargo, el espacio rgb es insensible a la orientación de la superfície, dirección de la iluminación e intensidad de la iluminación [66]. Similarmente, los canales H y S, que a continuación se definirán, son también invariantes para superfícies mates de color uniforme bajo iluminación blanca. Es por ello, por lo que en nuestro detector se han incluido los canales rgb normalizados y HSI(V), y se ha descartado el RGB. Aunque el canal I(V) se ha mantenido, en general, sus resultados de detección son bastante inferiores a los de los canales rgb, como se mostrará en el siguiente capítulo.

6.2.1.2. HSV y HSI

La representación de los colores en el espacio RGB es adecuada para el funcionamiento de los dispositivos electrónicos como monitores y cámaras, pero es difícil de interpretar intuitivamente por los humanos. Los espacios de color HSV (*Hue-Saturation-Value*) y HSI (*Hue-Saturation-Intensity*) proporcionan una representación más inteligible en cuanto al manejo del color.

El matiz (H) es un atributo del color que describe su pureza (amarillo puro, rojo, naranja, ...), mientras la saturación (S) proporciona una medida del grado en el que el color puro es diluido con luz blanca. La figura 6.21 ilustra el espacio de color HSV de tres formas diferentes: una representación sobre una rueda de color (fig.6.21(a)), donde el canal H se representa por una región circular y aparte, una región triangular se utiliza para representar S y V ; otra visualización en forma de cono (fig. 6.21(b)), donde H es una forma tridimensional cónica, S se representa por la distancia del centro a una sección interior del cono; o bien como un objeto cilíndrico (fig. 6.21(c)), donde H varía a lo largo de la circunferencia exterior del cilindro, S es la distancia desde el centro hasta una sección circular interna, y V varía de arriba a abajo.

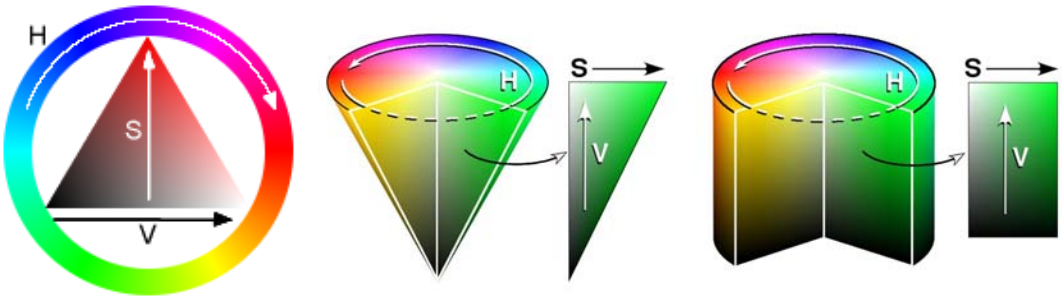
El modelo HSI se muestra en la figura 6.22 mediante su aproximación cónica. El modelo HSI es similar al HSV salvo que el canal V varía de negro a blanco en un único prisma, a diferencia del doble prisma del canal I [105]. La figura 6.23 ilustra esta diferencia entre los espacios de color HSV y HSI.

El canal I se definió en la ecuación 6.10. Las expresiones de los canales H, S y V en función de los valores R, G y B son las siguientes [66]:

$$H(R,G,B) = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right) \tag{6.12}$$

$$S(R,G,B) = 1 - \frac{\text{mín}(R,G,B)}{(R+G+B)} \tag{6.13}$$

$$V(R,G,B) = \text{máx}(R,G,B) \tag{6.14}$$



(a) (b) (c)

Figura 6.21: Modelo de color HSV: (a) representación sobre una rueda de color; (b) representación cónica.

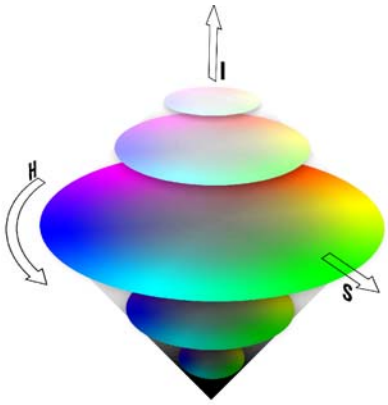


Figura 6.22: Modelo de color HSI.

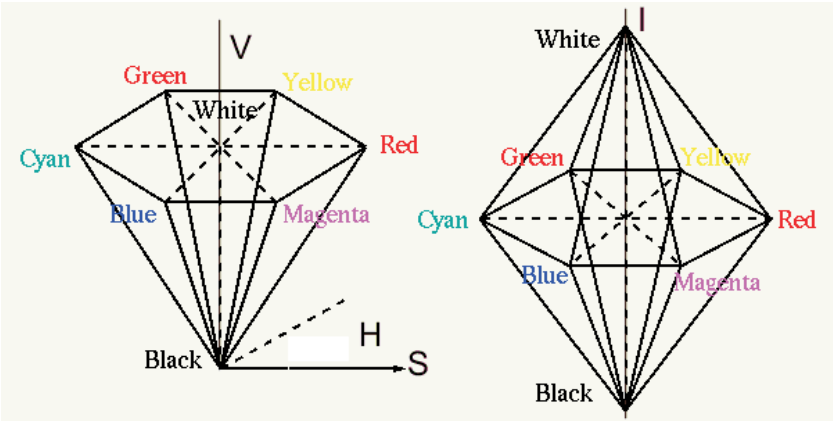


Figura 6.23: Diferencia entre los modelos de color HSV (izqda.) y HSI (dra).

6.2.1.3. $l_1l_2l_3$ y $c_1c_2c_3$

Los espacios de color $l_1l_2l_3$ y $c_1c_2c_3$ se definen también a partir de las tres matrices de color RGB mediante las siguientes transformaciones:

$$l_1 = \frac{(R-G)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \tag{6.15}$$

$$l_2 = \frac{(R-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \tag{6.16}$$

$$l_3 = \frac{(G-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \tag{6.17}$$

$$c_1 = \arctan\left(\frac{R}{\text{máx}\{G,B\}}\right) \tag{6.18}$$

$$c_2 = \arctan\left(\frac{G}{\text{máx}\{R,B\}}\right) \tag{6.19}$$

$$c_3 = \arctan\left(\frac{B}{\text{máx}\{R,G\}}\right) \tag{6.20}$$

Teóricamente, según el modelo de reflexión de [146], Gevers y Smeulders [66] demuestran que los espacios de color $l_1l_2l_3$ y $c_1c_2c_3$ son invariantes al punto de vista, la geometría del objeto y la iluminación. Sus resultados experimentales, mediante el uso de histogramas para realizar la clasificación de los objetos, confirman este hecho: en un entorno con iluminación blanca, el poder discriminativo de $l_1l_2l_3$, H, $c_1c_2c_3$ y rgb es aproximadamente el mismo. El canal S proporciona resultados ligeramente inferiores, mientras que RGB proporciona los peores resultados en el reconocimiento de objetos.

En la figura 6.24 aparece una imagen en cada uno de los canales de información de apariencia que incorpora nuestro detector.

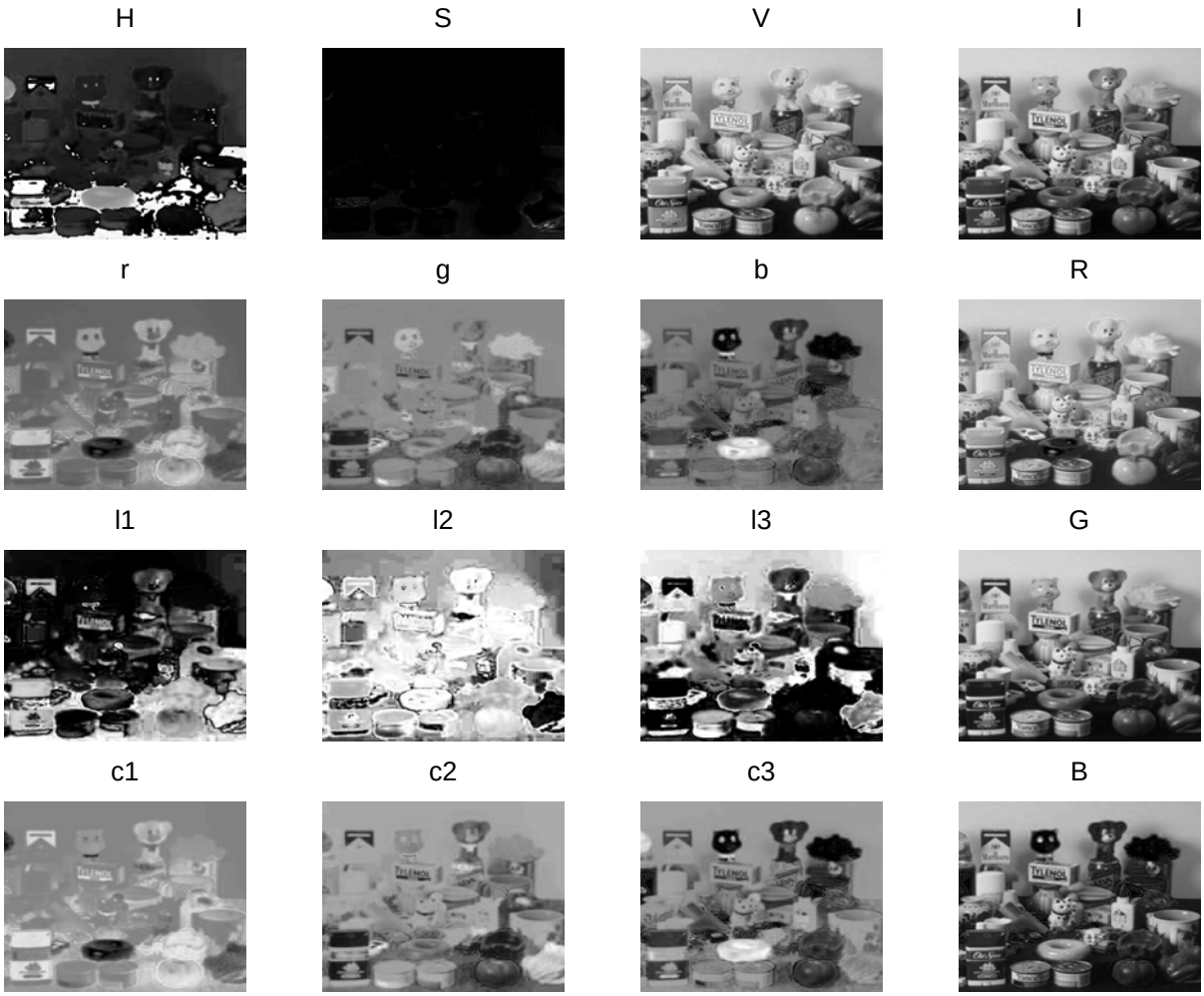


Figura 6.24: Espacios de color utilizados la etapa de detección multicanal PCA.

6.2.2. Canales de histogramas

Están formados por los ***histogramas***, en un rango de 256 niveles, de los fragmentos de las imágenes de entrenamiento en los espacios de color descritos en la sección anterior, concretamente: H, S, r, g, b, l1, l2, l3, c1, c2, c3 y la concatenación de RGB. A diferencia de los canales de información de apariencia, en los canales de histogramas se ha empleado la unión de los histogramas R, G y B, en vez de utilizar los canales I y V, que son directamente proporcionales a R, G y B.

En la figura 6.25 aparece la imagen representada en la anterior figura 6.24 en cada uno de los canales de histogramas que incorpora nuestro detector.

Dado un espacio de color, el histograma de color se obtiene mediante la discretización de los colores de la imagen contando el número de veces que cada que cada color aparece en la imagen.

El uso de los histogramas de color como representaciones de los objetos presentes en las imágenes para reconocimiento fue propuesto por Swain y Ballard en [154], junto con una medida de comparación entre histogramas, la *intersección de histogramas*, que permite comparar directamente el histograma de una imagen modelo, I, y el de una nueva imagen test, M :

$$\sum_{j=1}^{n} \text{mín}(I_j, M_j) \tag{6.21}$$

donde n es el número de barras del histograma.

En el método de detección propuesto no se utilizarán directamente los histogramas para reconocimiento, sino el resultado de su transformación PCA. Aunque el histograma de una imagen pueda ser dimensionalmente muy inferior al tamaño de la imagen en píxeles, en nuestro método resulta interesante aplicar la compresión PCA, puesto que el número de ejemplos de entrenamiento por objeto puede ser elevado al utilizar fragmentos de las imágenes de entrenamiento. Así, al utilizar PCA, no sólo se reduce la dimensión de los puntos en el espacio original, sino que también se obtiene la información contenida en las direcciones de máxima varianza de los datos, despreciando la información en direcciones que pueden ser ruido.

Los histogramas son invariantes ante traslación y rotación de las imágenes, y sólo cambian ligeramente si cambia la vista del objeto (si este es simétrico, evidentemente).

Los espacios de color rgb y $c_1c_2c_3$ no están definidos para el color negro (R=G=B=0), y el canal H y el espacio $l_1l_2l_3$ tampoco están definidos a lo largo del eje acromático (R=G=B). Además, estos espacios son muy inestables en estas singularidades, donde una pequeña perturbación en los valores RGB provocará un gran salto en los valores transfor-

mados. Para evitar las singularidades en esos casos límite se asigna el valor cero a cada una de las coordenadas de los espacios de color conflictivos, aunque ello suponga perturbar la información presente en la imagen. Este inconveniente no se puede resolver de manera sencilla cuando se utilizan las imágenes en estos espacios de color como canales de información para el detector (los canales de apariencia). Sin embargo, si es posible evitar las singularidades al utilizar los histogramas como canales de información. Una posible solución propuesta en [125] es, durante la construcción del histograma, considerar sólo los valores rgb cuyo valor de I sea superior a 30 (en un rango de 256 valores), y rechazar los valores de H si el valor S·(R+G*B) es menor que 9 (en un rango de 256 valores), puesto que H es inestable cuando S está cerca del 0.

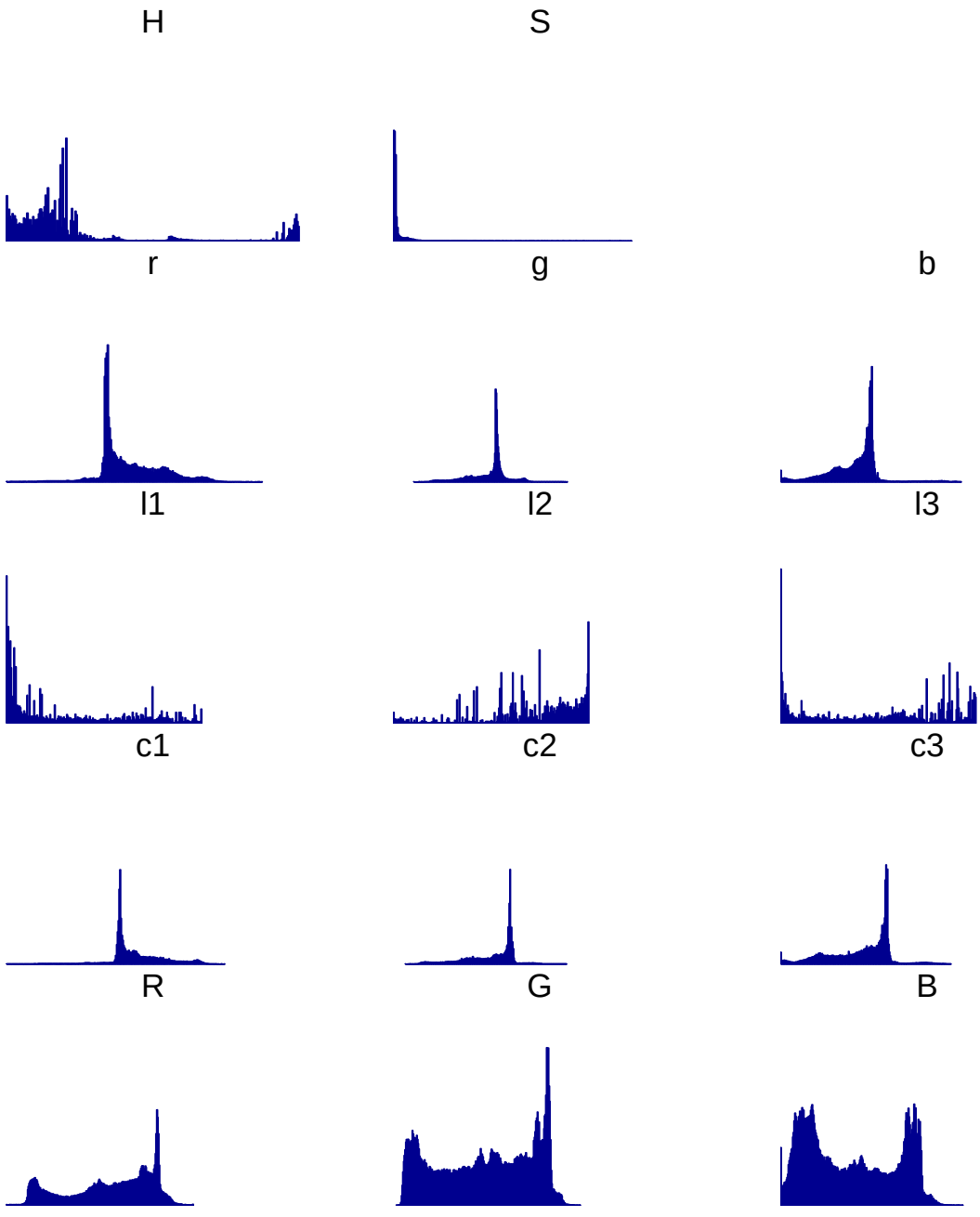


Figura 6.25: Histogramas utilizados la etapa de detección multicanal PCA.

6.2.3. Canales de forma

Los canales de forma de nuestro método están compuestos por fragmentos de las imágenes de bordes de las imágenes obtenidas a partir de los espacios de color antes expuestos. El detector de bordes utilizado es el popular detector de bordes de Canny [31]. La figura 6.26 muestra la *caja amarilla* en cada uno de los espacios de color anteriores, y la figura 6.27 muestra cada una de las imágenes de la *caja amarilla* tras realizar la extracción de bordes.

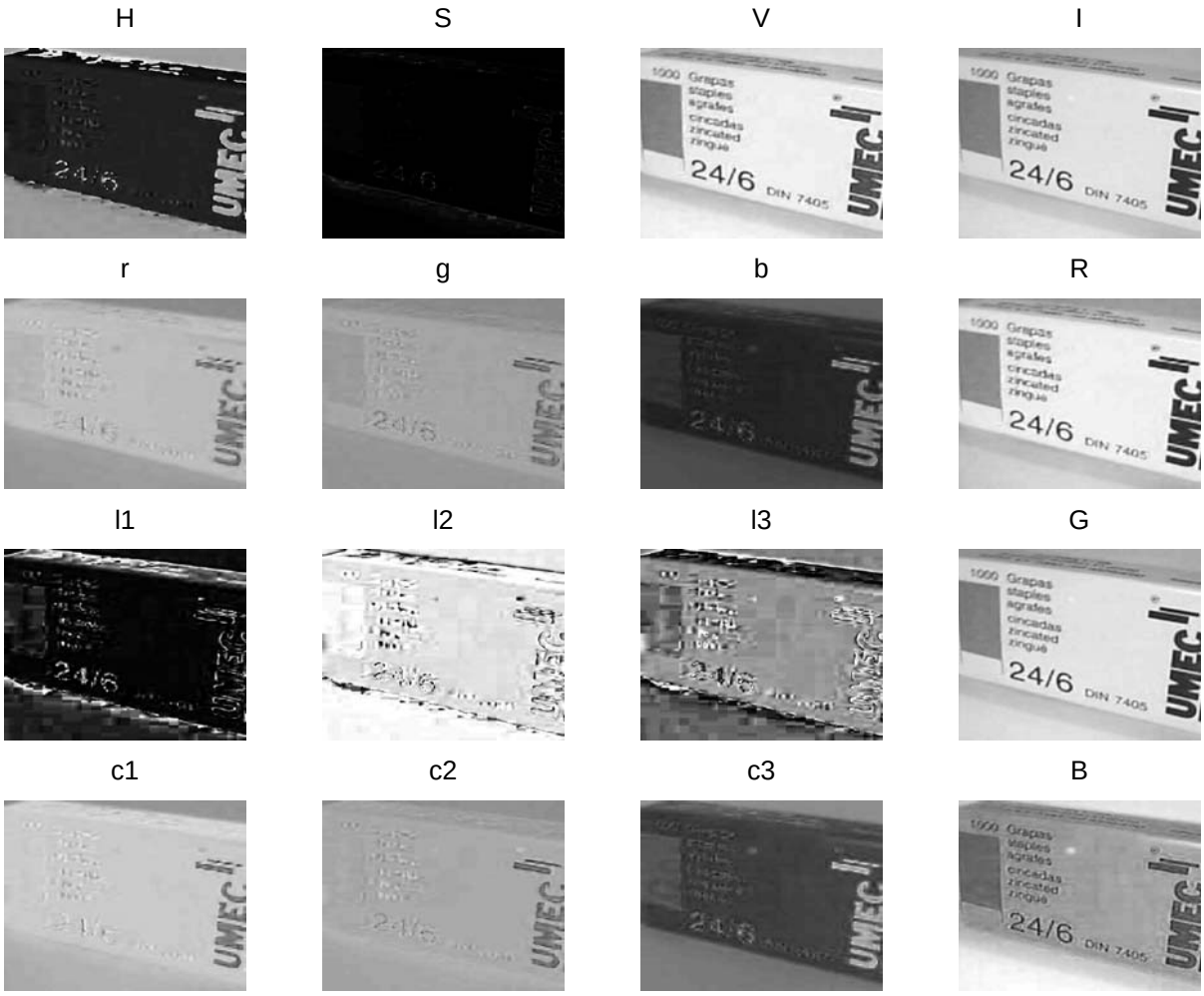


Figura 6.26: Imágenes de la *caja amarilla* en cada uno de los espacios de color utilizados la etapa de detección multicanal PCA.

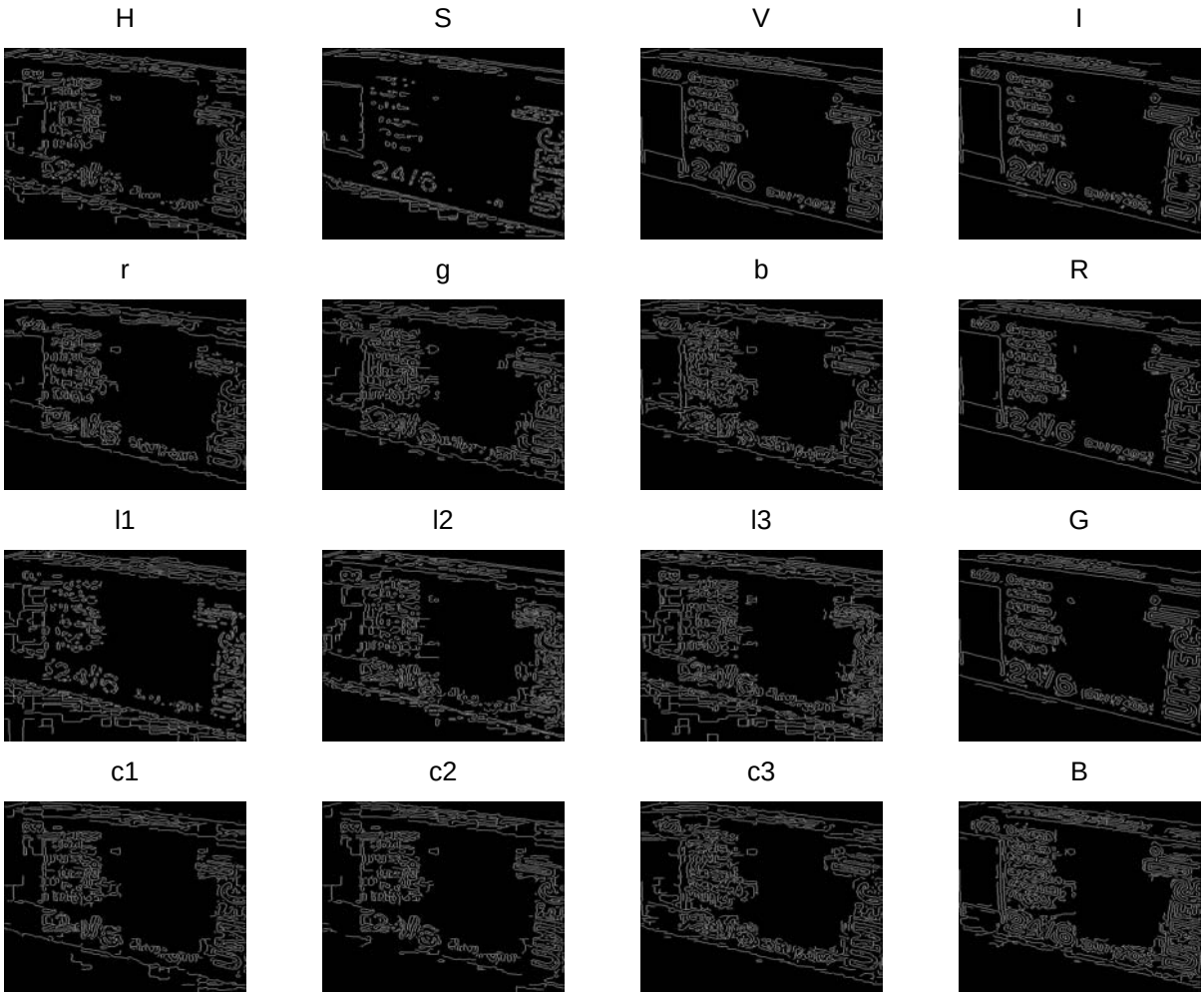


Figura 6.27: Imágenes de bordes de la *caja amarilla* en cada uno de canales de color anteriores.

6.2.4. Canal ON-OFF

En la sección 5.3 se presentaron los resultados obtenidos al realizar una comparativa del funcionamiento de PCA e ICA con diversos canales o filtrados de las imágenes. Este estudio indica que el uso de las imágenes filtradas por el canal ON-OFF y el operador LoG junto a la técnica de NMFSC produce mejores resultados en el reconocimiento en presencia de oclusiones. Este hecho, nos lleva a seleccionar el canal LoG_{ON-OFF} como el que se utilice dentro de nuestro sistema como canal de información de los objetos en la etapa de búsqueda multiescala. Para filtrar las imágenes a través de este canal, se utiliza la máscara del operador LoG descrita en la ecuación 5.24 y la definición de procesado ON-OFF:

$$\begin{aligned} \mathbf{Xp} &:= \text{máx}(\mathbf{X}, 0) \\ \mathbf{Xn} &:= -\,\text{mín}(\mathbf{X}, 0) \\ \mathbf{X}_{ON-OFF} &:= [\mathbf{Xp}; \mathbf{Xn}] \\ \mathbf{X}_{ON-OFF} &:= \text{máx}(\mathbf{X}_{ON-OFF}, 10^{-4}) \end{aligned} \tag{6.22}$$

6.3. Ajuste automático de la influencia de cada canal

Durante la etapa de detección multicanal PCA del objeto, la escena se divide en múltiples ventanas en cada uno de los canales de información utilizados, y estas ventanas transformadas mediante PCA se comparan con los modelos PCA del objeto obtenidos durante el entrenamiento. De esta manera, se genera un mapa de probabilidades de la escena, $\mathcal{M}_i$, por cada canal i, asociando a cada píxel de la escena un valor de probabilidad relativo a la existencia del objeto en esa zona de la escena. Se obtienen tantos mapas de probabilidades de escena, como canales de información se consideren. Posteriormente, se realiza una ponderación de estos mapas, para obtener un mapa de probabilidades global de la escena, $\mathcal{M}$, siendo la siguiente expresión la relación entre canales más sencilla a considerar:

$$\mathcal{M} = \frac{K_1 \cdot \mathcal{M}_1 + K_2 \cdot \mathcal{M}_2 + \ldots + K_N \cdot \mathcal{M}_N}{K_1 + K_2 + \ldots + K_N} = \frac{\sum_{i=1}^{N} K_i \cdot \mathcal{M}_i}{\sum_{i=1}^{N} K_i} \tag{6.23}$$

En la expresión anterior, los parámetros K_i indican la importancia de cada canal i para la construcción del mapa de probabilidades global.

En este apartado se discute cómo obtener estos parámetros automáticamente para la etapa de detección basada en PCA. Por otro lado, la obtención del valor de los pesos durante la etapa de detección basada en NMFSC, dependerá de la aplicación final del sistema de visión. En concreto, el peso que se asocie con cada escala de búsqueda dependerá de la información que se disponga acerca de la distancia entre el objeto a detectar y la cámara. En caso que esta distancia sea conocida con precisión, el proceso multiescala no será necesario y se utilizará la misma escala que la utilizada durante el entrenamiento. Sin embargo, si se desconoce totalmente la distancia a la que se encuentra el objeto, el proceso multiescala es necesario y en principio, la ponderación entre las escalas se realizaría por igual. Existirán situaciones intermedias en las que se conozca que la distancia al objeto se halle con gran probabilidad dentro de un cierto rango; en estos casos, los pesos se ajustarán de modo que en ese rango sean más elevados que en el resto de escalas, utilizando una expresión similar a la ecuación 6.23. Es importante tener en cuenta que el proceso multiescala tiene unos límites naturales, inherentes a la información visual que contiene la imagen de la escena, es decir, si el objeto está muy cerca de la cámara, o todo lo contrario, se encuentra muy alejado de ésta, ambos algoritmos serán incapaces de detectar la presencia del objeto, bien porque no se podrán obtener fragmentos con la suficiente información o bien porque no se podrán obtener las características locales propias del método NMFSC.

A continuación se describe la metodología propuesta para el ajuste automático de los pesos de los canales utilizados en la etapa de detección multicanal PCA.

6.3.1. Fiabilidad de cada canal de información

Los coeficientes K_i deben ser proporcionales a la fiabilidad de cada canal. Existen múltiples factores que afectan a tales fiabilidades relativas:

- *Tipo de objeto*: para ciertos objetos serán más fiables unos canales que otros. Por ejemplo, objetos con colores muy uniformes pueden ser distinguidos fácilmente a través de los canales de histogramas, mientras que objetos en los que los colores son más variables requieren que los pesos de los canales de color sean bajos para obtener buenos porcentajes de reconocimiento. Objetos con símbolos o texto podrán ser reconocidos más fácilmente utilizando los canales de forma.

- *Entorno que rodea al objeto*: en función del tipo de entorno, también puede variar la influencia relativa de cada canal. Fondos sencillos o incluso uniformes favorecen en general a todos los canales pero en particular a los basados en bordes. Dependiendo del tipo de uniformidad del fondo, se beneficia a ciertos canales: fondos uniformes en color benefician a los canales de histogramas; mientras que fondos uniformes en textura pueden beneficiar a otros canales.

- *Tipo de iluminación*: cuando la iluminación de la escena a reconocer es muy distinta de la utilizada durante el entrenamiento, los canales de apariencia en niveles de gris (relativos a I o V) se ven perjudicados.

La existencia de todos estos factores hace que el ajuste de pesos óptimo varíe en función de cada experimento. En cualquier caso, el factor más relevante es el tipo de objeto a detectar, con lo que en la presente tesis se propone utilizar una ponderación específica de canales para cada objeto.

6.3.2. Propuesta de aprendizaje de pesos mediante entrenamiento

Se propone asignar los pesos de cada canal mediante un procedimiento de aprendizaje a partir de ejemplos de entrenamiento. Los ejemplos son escenas similares a las que el sistema encontrará durante el funcionamiento automático, en las que el objeto de interés se encuentra segmentado y etiquetado. Ejemplos de escenas etiquetadas con sus correspondientes escenas originales se muestran en las figuras 6.28 y 6.29.

En función de la variabilidad previsible en las imágenes que encontrará el sistema en funcionamiento automático, los ejemplos de entrenamiento deberán cubrir en la medida de lo posible tal variabilidad (diversas condiciones de iluminación, distintos fondos, distintas distancias entre el objeto de interés y la cámara, etc.)

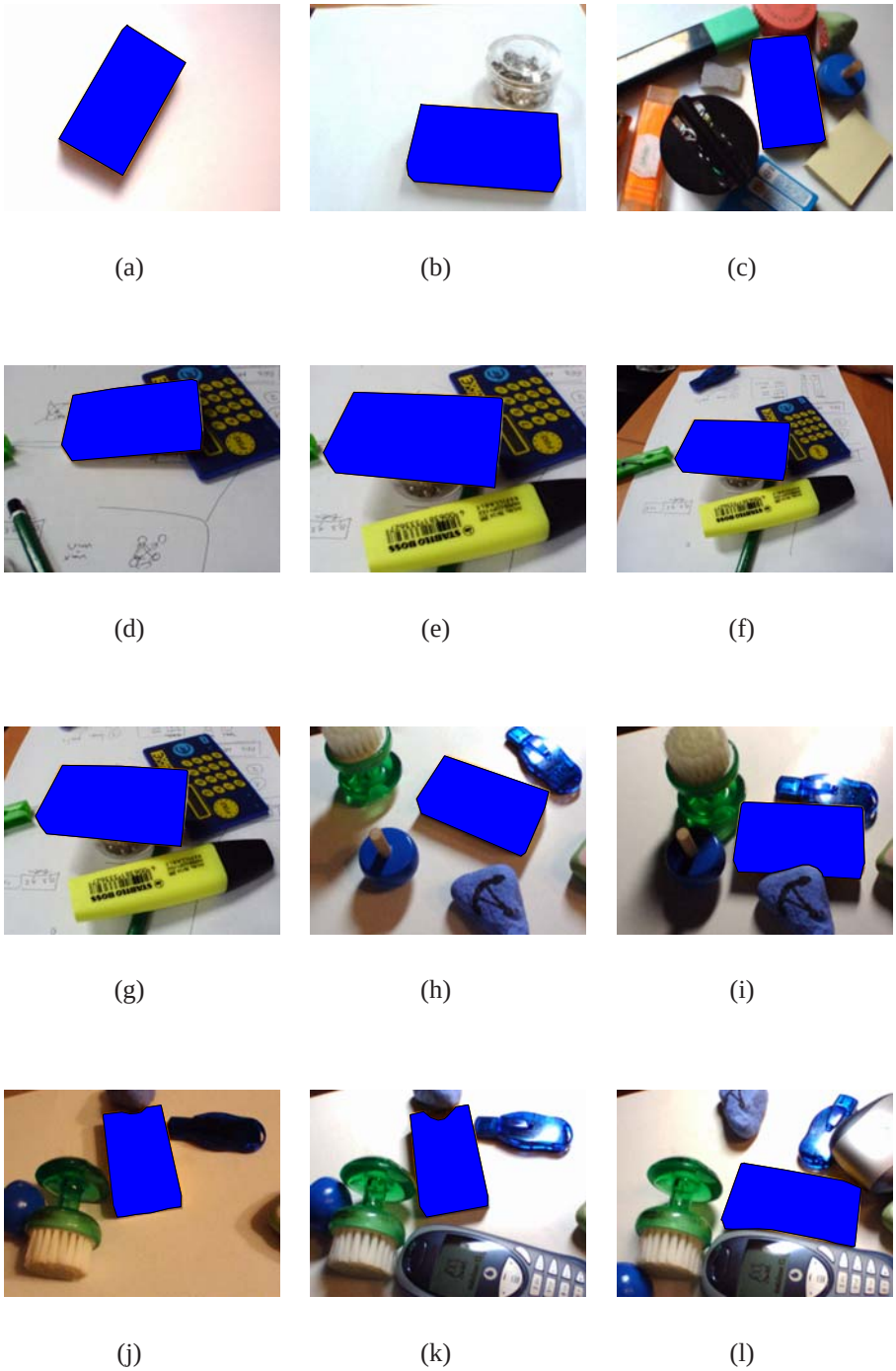
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figura 6.28: Escenas originales.



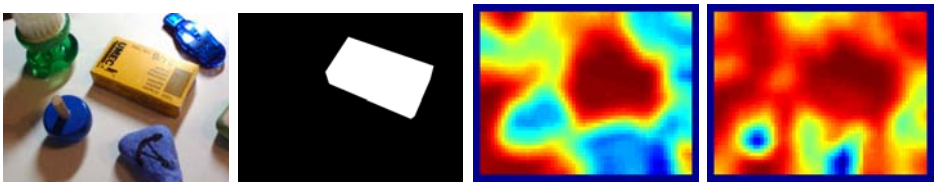
(a) (b) (c)

(d) (e) (f)

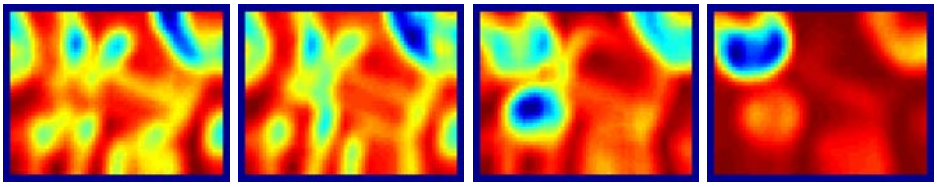
(g) (h) (i)

(j) (k) (l)

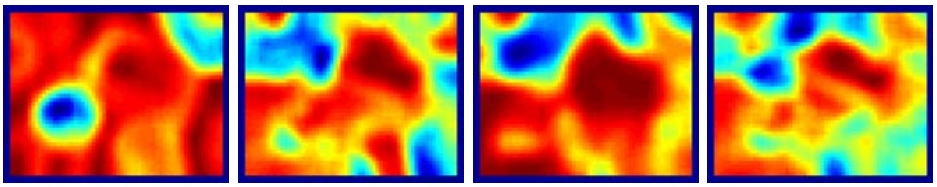
Figura 6.29: Escenas etiquetadas.



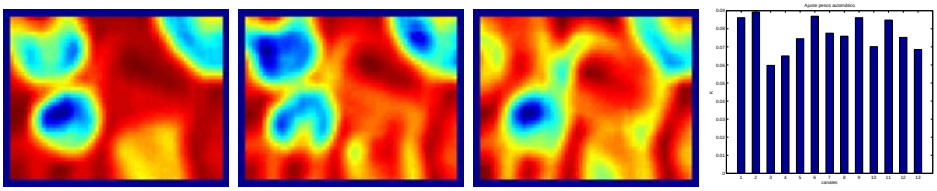
(a) Escena (b) Etiquetada (c) $H, 0{,}0861$ (d) $S, 0{,}0893$



(e) $V, 0{,}0597$ (f) $I, 0{,}0650$ (g) $r, 0{,}0745$ (h) $g, 0{,}0870$



(i) $b, 0{,}0776$ (j) $l_1, 0{,}0759$ (k) $l_2, 0{,}0861$ (l) $l_3, 0{,}0701$



(m) $c_1, 0{,}0848$ (n) $c_2, 0{,}0752$ (ñ) $c_3, 0{,}0685$ (o) Pesos

Figura 6.30: Ejemplo sobre el ajuste automático de pesos de los canales: (a) Escena utilizada para el cálculo de los pesos; (b) Escena etiquetada; (c)-(ñ) Canales de apariencia, junto con su correlación normalizada respecto de la escena etiquetada;(o) Pesos de cada canal, que en este caso, al utilizar sólo una escena de entrenamiento, son iguales a la correlación de cada canal.

La idea básica consiste en considerar que la influencia de cada canal ha de ser proporcional a la correlación que existe entre el mapa de probabilidad de dicho canal y la imagen de la escena etiquetada (en la que las zonas donde se encuentra el objeto tienen valor 1 y el fondo de la imagen tiene valor 0). Para calcular correctamente la correlacion, los mapas de probabilidad de cada canal han de estar normalizados según el método descrito en el apartado 6.1.1.2, es decir, el píxel de máxima distancia al modelo tendrá un valor de probabilidad 0 y el píxel de mínima distancia, un valor 1. La figura 6.30 muestra algunos ejemplos de los resultados ofrecidos por diversos canales y su correlación con una de las escenas etiquetadas, en concreto la representada en la figura 6.28(h).

En resumen, el procedimiento a utilizar para calcular la influencia óptima de cada canal es el siguiente:

Para cada objeto:

1. Recopilar escenas de entrenamiento y etiquetarlas.
2. Calcular mapas de distancia, $\mathcal{D}_i$, en cada uno de los canales y escalarlos entre 0 y 1 linealmente (mapas de probabilidad, $\mathcal{M}_i$).
3. Calcular la correlación para cada escena y cada canal; a tales correlaciones se les llamará c_{ij} donde j hace referencia a la escena (se supone un total de T escenas) e i, al canal (se supone un total de N canales).
4. Calcular la fiabilidad o el peso de cada canal como la media de las correlaciones entre todas las escenas de entrenamiento:

$$K_i = \frac{1}{T} \sum_{j=1}^{T} c_{ij} \tag{6.24}$$

A través del ejemplo de la figura 6.30 puede apreciarse cómo los valores de correlación permiten asignar pesos más elevados a los canales que mejor detectan un cierto objeto, pero las diferencias de peso entre los distintos canales no son muy acusadas, a pesar de que los comportamientos de los mismos sí son muy distintos (basta comparar la imagen etiquetada con cada uno de los mapas de probabilidad de los canales).

En el algoritmo desarrollado en la presente tesis se ha incorporado un parámetro que permite que las diferencias entre los canales se hagan más acusadas si el usuario lo desea. Este parámetro modifica los mapas de probabilidad utilizados para el cálculo de la correlación, aplicándoles un escalado no lineal. La función utilizada se indica en la ecuación 6.25, donde γ es el *factor de escalado* configurable y $\tilde{\mathcal{M}}_i$ el mapa de probabilidad

transformado:

$$\tilde{\mathcal{M}}_i = \frac{e^{\gamma \cdot \mathcal{M}_i} - 1}{e^{\gamma} - 1} \tag{6.25}$$

La figura 6.31 muestra el escalado resultante en función del valor del parámetro γ.

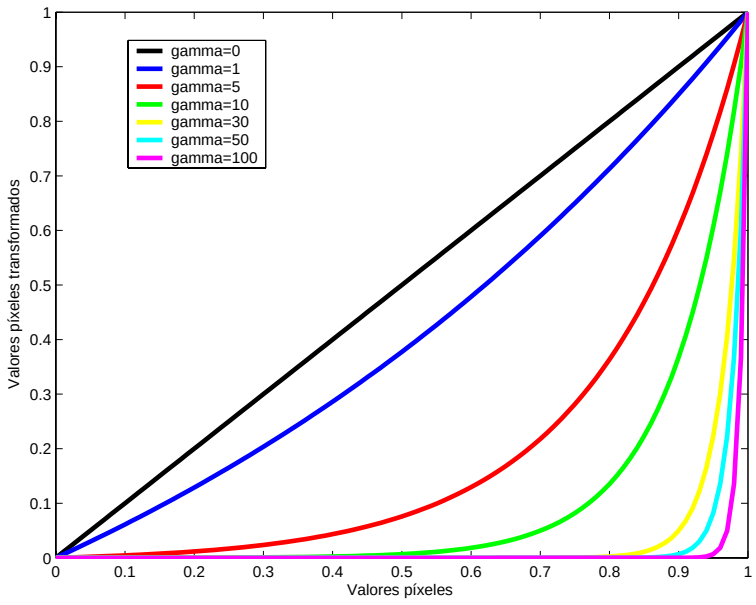
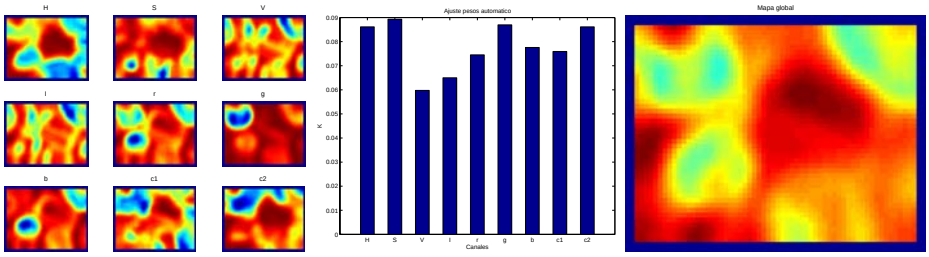


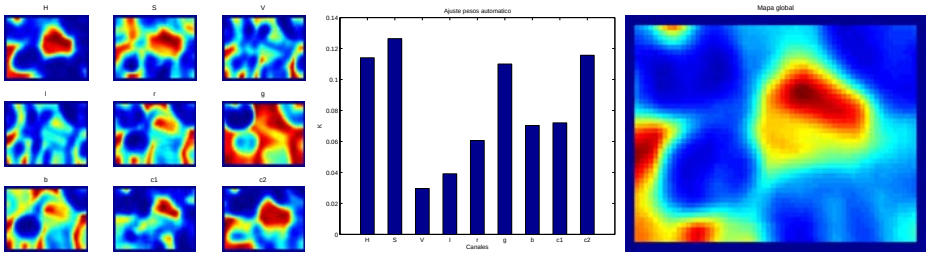
Figura 6.31: Transformación en los valores de los píxeles de los nuevos mapas de probabilidad $\tilde{\mathcal{M}}_i$.

Para comprobar el efecto del factor de escalado sobre los pesos que se utilicen para los diferentes canales, se ha realizado un experimento[7] en el que se calculan las ponderaciones de los canales para distintos valores de γ. Los resultados se muestran en la figura 6.32. Se puede apreciar cómo las ponderaciones son más extremas a medida que aumenta el valor de γ, (e incluso algunos de los canales obtienen ponderación aproximadamente igual a 0), y cómo esto redunda en una mejor segmentación de la región donde se encuentra el objeto.

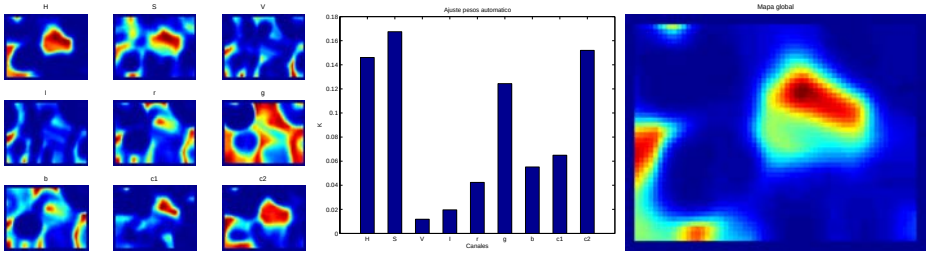
[7] Utilizando la misma escena de entrenamiento que en el ejemplo de la figura 6.30.



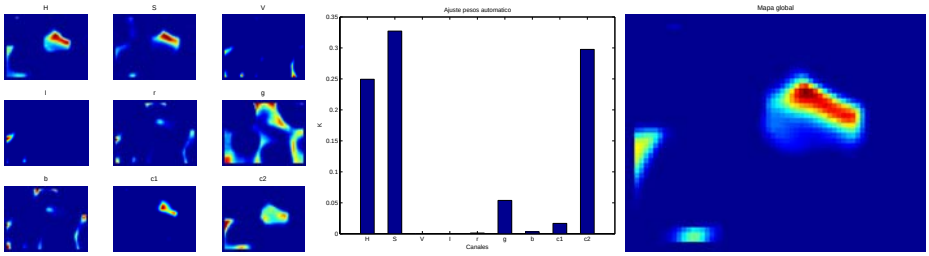
(a) $\gamma = 0$ (b) Pesos (c) Mapa global



(d) $\gamma = 5$ (e) Pesos (f) Mapa global



(g) $\gamma = 10$ (h) Pesos (i) Mapa global



(j) $\gamma = 50$ (k) Pesos (l) Mapa global

Figura 6.32: Efecto del escalado no lineal en los mapas de probabilidad sobre la ponderación de los canales.

En la figura 6.32 la columna de la izquierda muestra 9 mapas de probabilidad relativos a los canales de apariencia H, S, V, I, r, g, b, c1 y c2 con transformaciones distintas. En la fila superior, los mapas de probabilidad están normalizados linealmente entre 0 y 1; mientras que las filas inferiores utilizan la transformación definida en la ecuación 6.25 con valores para el parámetro γ, 5, 10 y 50. En la columna central, se muestra la ponderación de canales obtenida y en la columna derecha, aparecen los mapas de probabilidad globales para cada uno de los factores de escalado, obtenidos utilizando la expresión de ponderación entre canales (ecuación 6.23).

Lógicamente, el funcionamiento del sistema de ajuste de pesos de los canales propuesto será mejor cuanto mayor sea el número de escenas etiquetadas disponibles para el entrenamiento. En los experimentos realizados en la presente tesis, se ha comprobado que 10 escenas etiquetadas son suficientes para un funcionamiento correcto, si su elección es adecuada (las escenas cubren la mayor variabilidad posible).

Como experimento adicional para comprobar que un número de 10 escenas es suficiente, el gráfico de la figura 6.33 muestra la evolución de los coeficientes K_i con el número de escenas etiquetadas utilizado. Es posible apreciar cómo los coeficientes se mantienen aproximadamente uniformes a partir de 8-10 escenas, y el resto de datos de entrenamiento suministrados sólo sirven para realizar pequeños ajustes. Los resultados de la figura 6.33 corresponden a la evolución de los pesos de los 13 canales de apariencia para la *caja amarilla*, utilizando hasta 25 escenas etiquetadas, estando entre ellas, las escenas mostradas en las figuras 6.28 y 6.29.

El gráfico anterior demuestra cómo el sistema propuesto para el ajuste de pesos es válido, y no es necesario utilizar un gran número de escenas etiquetadas (lo cual habría dificultado la aplicación en un entorno real).

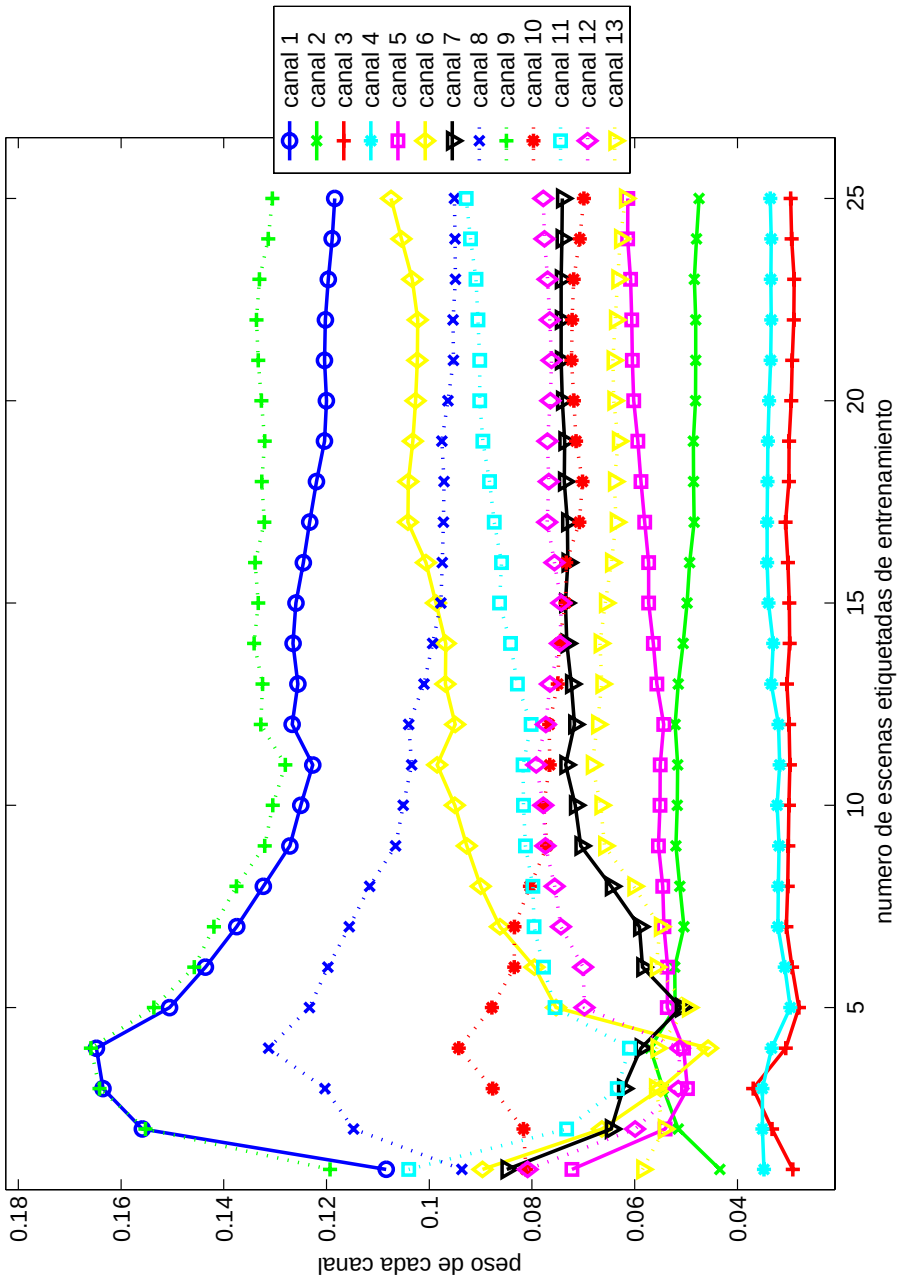


Figura 6.33: Evolución del vector de pesos al añadir imágenes etiquetadas de entrenamiento.

6.4. Umbral automático de reconocimiento

6.4.1. Necesidad de establecer un umbral

El resultado de los dos algoritmos de detección de objetos propuestos en la presente tesis se representa mediante un mapa de probabilidades, que indica, para cada píxel de la imagen, la probabilidad de que el objeto a detectar ocupe tal píxel. Un ejemplo de mapa de probabilidades se muestra en la figura 6.34 donde el objeto a detectar es *la caja amarilla* y las zonas del mapa con probabilidades más cercanas a la unidad servirán para localizar tal caja. El mapa mostrado puede representar el resultado de uno de los canales (PCA) o una de las escalas (NMFSC) por separado, o bien el resultado combinado de todos los canales o escalas.

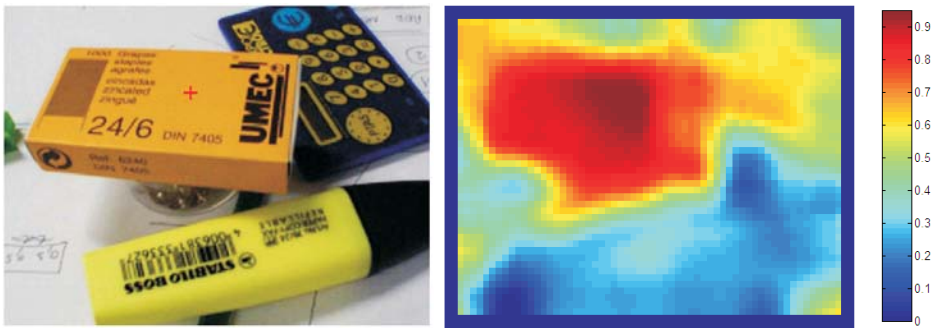


Figura 6.34: El mapa de probabilidades indica para cada píxel de la escena la probabilidad de que el objeto a detectar ocupe tal píxel.

En general, la probabilidad será tanto más cercana a la unidad cuanto más similar sea la escena a alguna de la imágenes (o los fragmentos) de entrenamiento. Se puede considerar que los valores más altos de probabilidad dentro del rango indican la presencia del objeto, aunque estos valores se encuentren alejados de la unidad. Si el único objetivo es detectar la posición del objeto en la imagen, basta con marcar como posición del mismo el píxel de mayor probabilidad (indicado con una cruz en la figura). Sin embargo, esta estrategia presenta algunos inconvenientes:

- No permite obtener una segmentación, aunque sea aproximada, del objeto.
- No permite determinar si el objeto está presente o no en la escena.
- El método de combinación de los resultados de la detección multicanal PCA y la detección multiescala NMFSC propuesto en esta tesis está basado en el establecimiento de un umbral.

Estos inconvenientes se pueden solventar con el establecimiento de un umbral, de modo que todos los píxeles con valor de probabilidad superior al umbral sean considerados

pertenecientes al objeto a detectar, tal y como se muestra en la figura 6.35. Si ninguno de los píxeles supera el umbral se concluye que el objeto buscado no está presente en la escena. Prácticamente cualquier aplicación que haga uso de los desarrollos de la presente tesis requirirá el establecimiento de un umbral. Sería inviable un método de detección de objetos que no descartase las imágenes en las que el objeto no está presente.

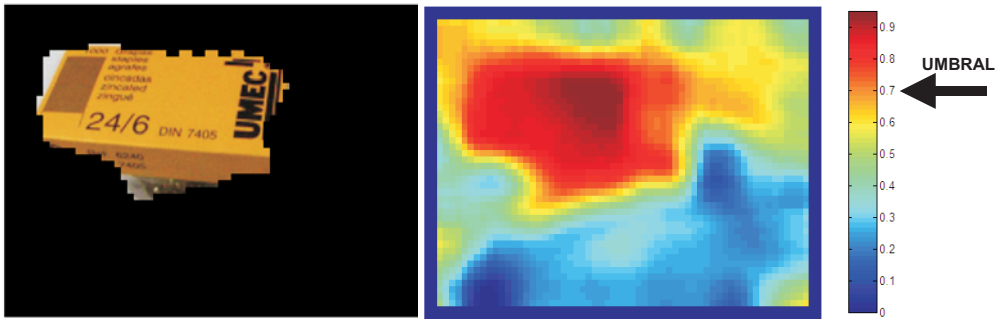
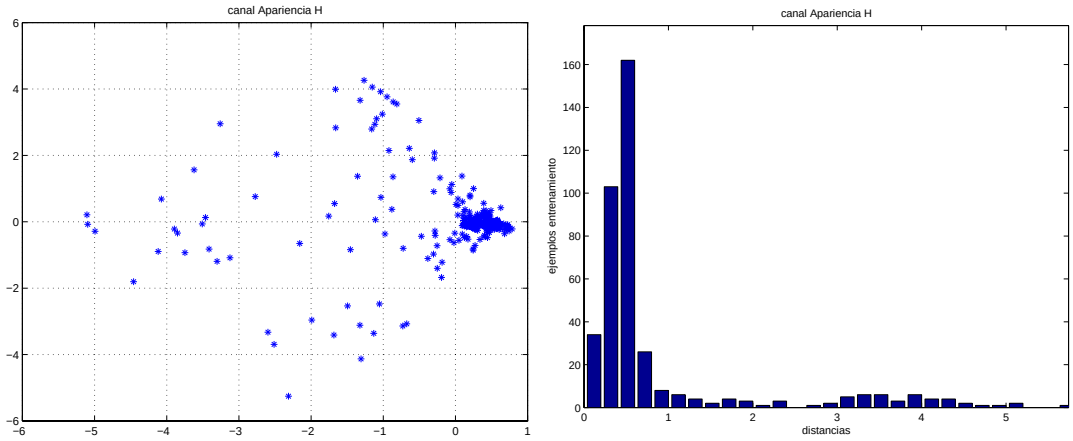


Figura 6.35: Es necesario el establecimiento de un umbral para la correcta detección del objeto.

6.4.2. Establecimiento del umbral a través de la distribución espacial de los modelos de apariencia

En este apartado se propone una metodología para el establecimiento automático del umbral de detección basada en la distribución espacial de los modelos de apariencia del objeto a detectar.

Como ya se vio anteriormente, los modelos de apariencia de los objetos, en los distintos canales y para ambos algoritmos, están formados por el conjunto de las proyecciones de las imágenes de entrenamiento, o bien por sus fragmentos. La figura 6.7(b) mostraba un ejemplo de modelo de apariencia de *la caja amarilla* en el canal H, utilizando sólo las tres primeras componentes principales. En tal figura se puede apreciar cómo la mayor parte de los puntos en el espacio tridimensional (cada punto representa la compresión mediante PCA de la imagen de un fragmento en un canal concreto) se agrupan en una pequeña región del espacio; mientras que algunos puntos se encuentran más alejados. Esto es lógico dado que cada imagen cubre un fragmento y una orientación particular del objeto, que en algunos casos pueden ser muy distintos entre sí. Las distancias existentes entre los fragmentos de entrenamiento se pueden representar en forma de histograma. Sobre un ejemplo concreto, un modelo de apariencia en el canal H de *la caja amarilla*, se calculan las distancias entre cada una de las imágenes de entrenamiento (proyecciones en el subespacio PCA) y su valor medio, y se genera un histograma con ellas. El resultado se muestra en la figura 6.36, donde se han utilizado sólo dos dimensiones espaciales tanto



(a) Proyecciones (b) Histograma

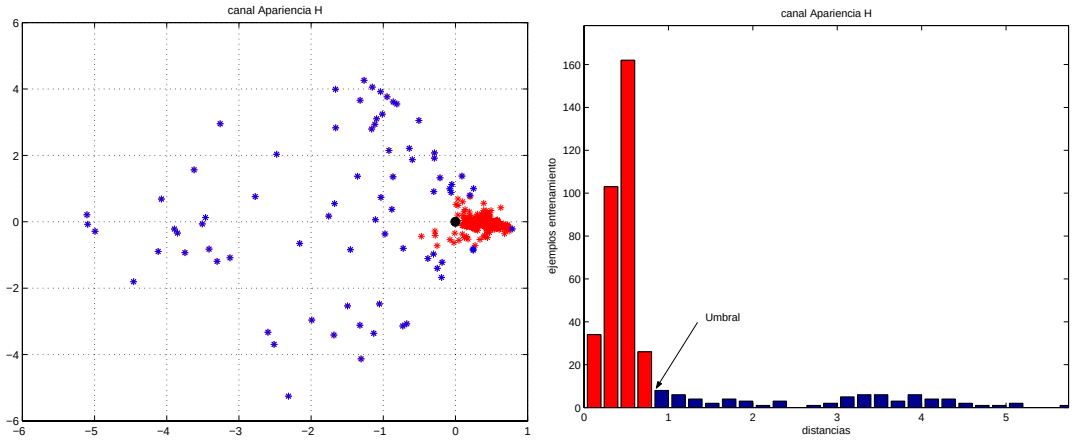
Figura 6.36: Las distancias existentes entre los fragmentos de entrenamiento se pueden representar en forma de histograma.

para la visualización de las proyecciones como para el cálculo de las distancias respecto del valor central[8].

Se puede considerar que el histograma mostrado es una representación de la distribución espacial del modelo de apariencia en un cierto canal. Esta distribución puede utilizarse para determinar de modo automático un umbral de reconocimiento. Se elige como umbral aquel valor de distancia que permite cubrir un cierto porcentaje del área del histograma. Si, por ejemplo, se fija un límite del $80\,\%$, los resultados sobre el ejemplo anterior se muestran en la figura 6.37, donde se ha utilizado el color rojo para marcar los puntos por debajo del umbral de distancia y el color azul para marcar los puntos por encima del umbral. Tanto sobre el gráfico bidimensional como sobre el histograma, los puntos rojos se corresponden con imágenes lo suficientemente próximas a la media como para ser aceptadas; los puntos azules indican exceso de alejamiento de la media.

La elección de un límite del $80\,\%$ puede parecer a priori arbitraria; sin embargo, este límite está relacionado con la mayor o menor simetría y uniformidad del objeto, es decir, de si muestra o no la misma apariencia en todas sus posibles orientaciones a la cámara y en todos sus posibles fragmentos. Un objeto que presente mucha variabilidad en sus vistas y poca uniformidad en su apariencia dará lugar a un modelo de apariencia donde las distancias de las proyecciones a la media son mayores que las proporcionadas por un objeto más simétrico y uniforme. Este hecho puede ser útil a la hora de fijar este límite

[8]En todos los casos el valor medio es 0 puesto que las proyecciones están centradas siguiendo la metodología de la técnica PCA.



(a) Proyecciones (b) Histograma

Figura 6.37: La distribución espacial del modelo de apariencia en un cierto canal puede utilizarse para determinar de modo automático un umbral de reconocimiento.

para la obtención del umbral automático, aunque en cualquier caso este parámetro es siempre ajustable por el usuario y permitirá que el sistema funcione con mayor o menor permisividad de detección.

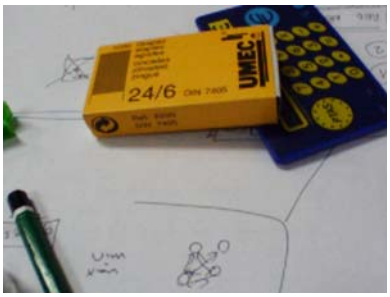
Para cada objeto y para cada canal de características se obtienen distribuciones espaciales distintas. Como ejemplo, en el anexo C.2 se muestran las distribuciones espaciales obtenidas para la *caja amarilla* en todos los canales, utilizando el límite del $80\,\%$ para el cálculo automático del umbral.

En los experimentos mostrados en el siguiente capítulo de la tesis, se ha establecido como límite para el cálculo del umbral automático, según esta metodología, un valor del $80\,\%$ para la mayoría de los casos. Sin embargo, con algunos objetos poco simétricos, se han utilizado valores ligeramente inferiores (hasta un $70\,\%$). No obstante, se ha comprobado que la elección de este parámetro no es crítica y que pequeñas variaciones del mismo (en un rango entre el $70\,\%$ y $90\,\%$) no deterioran apreciablemente el funcionamiento de la detección.

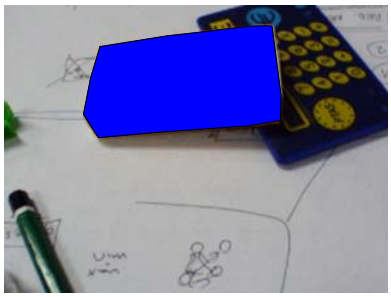
6.4.3. Establecimiento del umbral mediante aprendizaje con escenas etiquetadas

Como alternativa al cálculo automático del umbral mediante el estudio de la distribución de los modelos de apariencia, se propone la utilización de escenas etiquetadas. Tales escenas se utilizan previamente en el algoritmo para el ajuste automático de los pesos de cada canal, y por tanto están disponibles sin necesidad de realizar etiquetados adicionales.

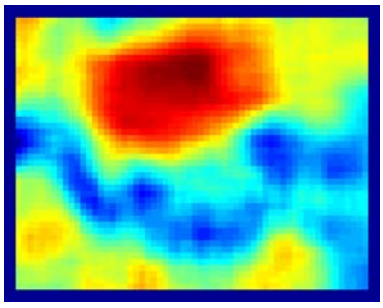
El procedimiento para la obtención del umbral se basa en optimizar el funcionamiento del sistema sobre las imágenes de entrenamiento etiquetadas. La figura 6.38 muestra una escena de entrenamiento junto con su etiquetado y el resultado de calcular la distancia respecto del modelo del objeto de interés, es decir su mapa de probabilidad.



(a) Escena



(b) Escena etiquetada



(c) Mapa probabilidad de la escena

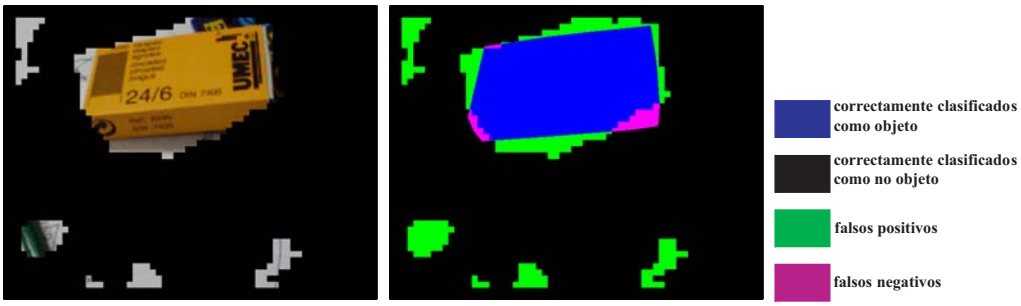
Figura 6.38: Escena original, escena etiquetada y mapa de probabilidad de la escena.

Si se establece un umbral de reconocimiento determinado, en base a la metodología anterior basada en la distribución espacial de los modelos, es posible marcar los píxeles donde se espera o no encontrar al objeto. Tales zonas se muestra en la figura 6.39(a).

Comparándolas con el área que realmente ocupa el objeto, se pueden encontrar cuatro tipos de píxeles distintos:

- correctamente clasificados como objeto
- correctamente clasificados como no objeto
- incorrectamente clasificados como objeto (*falsos positivos*)
- incorrectamente clasificados como no objeto (*falsos negativos*)

Todos estos tipos de píxeles, se muestran en la figura 6.39(b).



(a) Escena umbralizada (b) Falsos positivos y falsos negativos (c) Leyenda

Figura 6.39: (a) Al establecer un umbral de reconocimiento (0.57, en este ejemplo), es posible segmentar el objeto de la escena; (b) tipos de píxeles que aparecen en la escena umbralizada; (c) leyenda de los tipos de píxeles.

La modificación del umbral altera las zonas obtenidas: una elevación del umbral reduce el número de falsos positivos y aumenta el de falsos negativos; y una disminución del umbral produce el efecto contrario.

Si se representan en un gráfico tanto la variación del porcentaje de falsos positivos y negativos como la variación del porcentaje total de errores (suma de los anteriores) en función del umbral, se obtiene un resultado como el de la figura 6.40.

En dicha figura puede apreciarse cómo existe un valor del umbral para el cual el porcentaje total de píxeles erróneos es mínimo. En principio, este sería el úmbral óptimo a buscar, pero dependiendo de la aplicación puede interesar un umbral distinto: en ciertas aplicaciones puede interesar que el número de falsos positivos sea lo menor posible aunque a consecuencia de ello el número de falsos negativos aumente (por ejemplo, en una

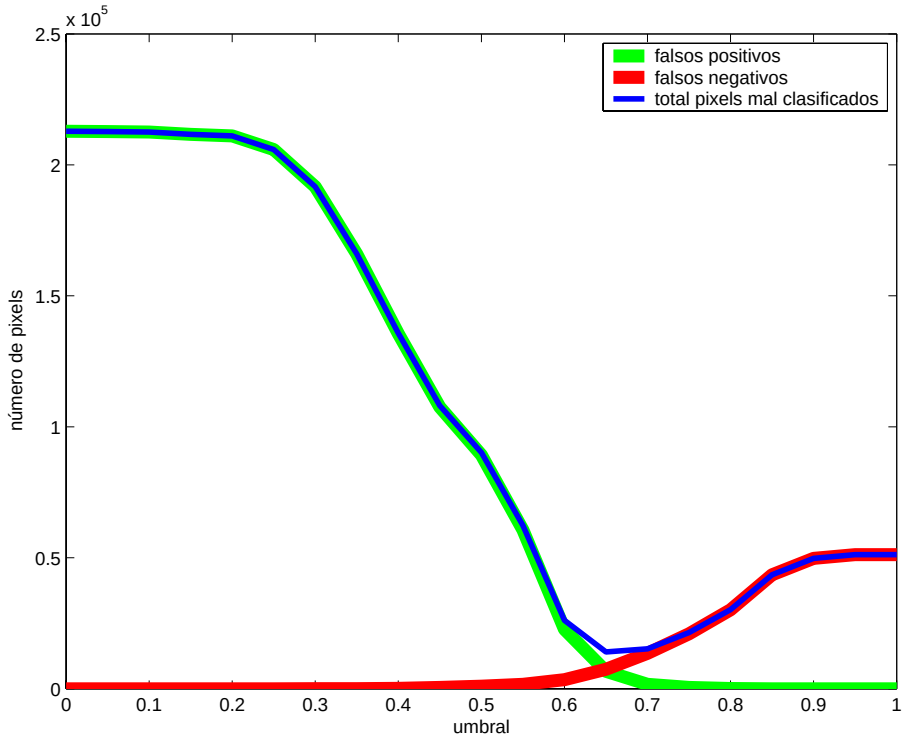


Figura 6.40: Variación de los porcentajes de error en función del umbral.

aplicación de reconocimiento de caras en la que lo más importante sea no aceptar a intrusos, por motivos de seguridad), mientras que otras aplicaciones pueden tener intereses contrarios.

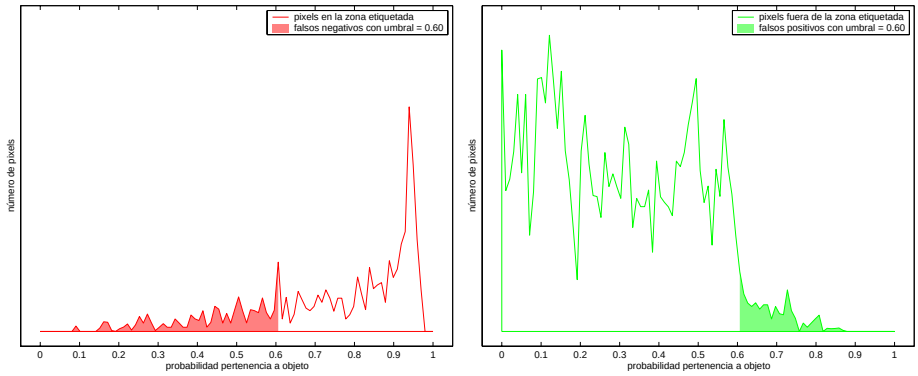
La propuesta del cálculo del umbral a partir de escena etiquetadas permite contemplar cualquier posibilidad como la anteriormente mencionada; basta con elegir el umbral en función de uno de los dos criterios siguientes:

- mínimo número total de errores en la clasificación esperable.
- relación porcentual entre el número de falsos positivos y el número de falsos negativos esperables. (Por ejemplo, una aplicación donde lo importante sea evitar los falsos positivos, puede fijarse una relación $25\,\% - 75\,\%$ entre el número total de falsos positivos y el número total de falsos negativos).

El procedimiento a utilizar para el cálculo automático del umbral de acuerdo con la propuesta realizada en este apartado es muy sencillo, se puede resumir en los siguientes pasos:

1. Recopilación de escenas de entrenamiento etiquetadas.
2. Cálculo de distancias al objeto de interés para cada píxel de cada imagen, es decir, obtención del mapa de probabilidad.
3. Creación de histogramas (en número de píxeles) para las distancias al objeto de interés (los valores de probabilidad) tanto de los píxeles donde se encuentra el objeto como de los píxeles donde no se encuentra el objeto.
4. Ajuste del umbral a partir de la superposición de histogramas.

Para explicar el procedimiento anterior, supóngase que el cálculo de distancias al objeto de interés para los píxeles donde se encuentra el objeto (etiquetados con un "1" en las escenas utilizadas durante el entrenamiento) se corresponde con el histograma de la figura 6.41(a) y el mismo cálculo para los píxeles donde no se halla el objeto (etiquetados con un "0") se corresponde con el histograma de la figura 6.41(b).



(a) (b)

Figura 6.41: Histogramas de probabilidad de pertenencia al objeto. (a) Píxeles pertenecientes a la zona etiquetada como objeto. (b) Píxeles pertenecientes a la zona etiquetada como fondo.

Lógicamente, el histograma a) se encuentra desplazado a la derecha (probabilidades altas) y el histograma b) se encuentra desplazado a la izquierda (probabilidades bajas). En función del umbral elegido, se puede visualizar el número de falsos positivos y falsos negativos sobre los histogramas, por separado como en la figura anterior o conjuntamente como en la figura 6.42.

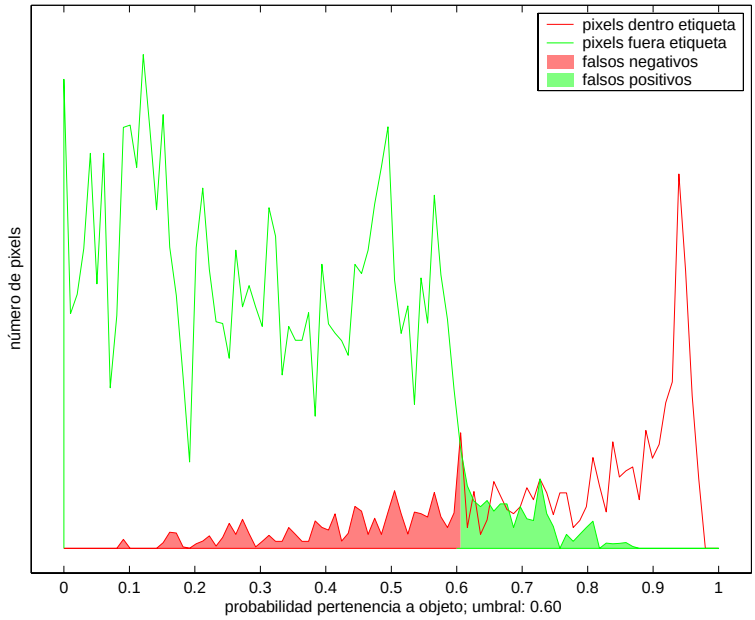
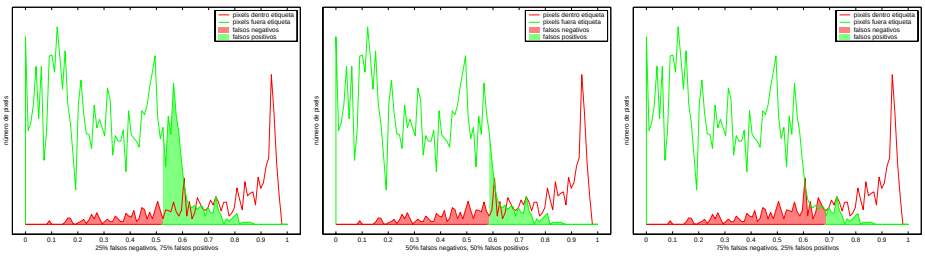


Figura 6.42: Combinación de los histogramas de píxeles etiquetados como objeto y como fondo.

El umbral se puede ajustar en función de lo deseado, tal y como se muestra en la figura 6.43. En general, lo razonable será o bien reducir el número total de píxeles mal clasificados o bien igualar el porcentaje esperado de falsos positivos y falsos negativos. No obstante en algunos casos puede interesar reducir en mayor medida el número de falsos positivos o el número de falsos negativos. La figura muestra los umbrales que habría que utilizar para conseguir (de acuerdo con las escenas de entrenamiento) unos porcentajes de $25-75\,\%$, $50-50\,\%$ y $75-25\,\%$.



(a) (b) (c)

Figura 6.43: Distintas selecciones del umbral en función del porcentaje de falsos positivos y falsos negativos deseado.

En la presente tesis, los experimentos se han realizado buscando limitar el número de falsos negativos. El número de falsos positivos es menos importante, dado que la intersección de escenas umbralizadas mediante el uso de los dos algoritmos elimina falsos positivos pero mantiene los falsos negativos.

Para apreciar claramente el efecto de la variación del umbral sobre los resultados esperables en el reconocimiento de objetos, se muestran en la figura 6.44 seis valores distintos para el umbral y su efecto sobre los histogramas; y en la figura 6.45 se refleja el resultado que se obtendría en términos de falsos positivos y falsos negativos sobre la escena de entrenamiento. En esta última figura se utiliza el código de colores habitual (ver figura 6.39(c)) para los falsos positivos y falsos negativos.

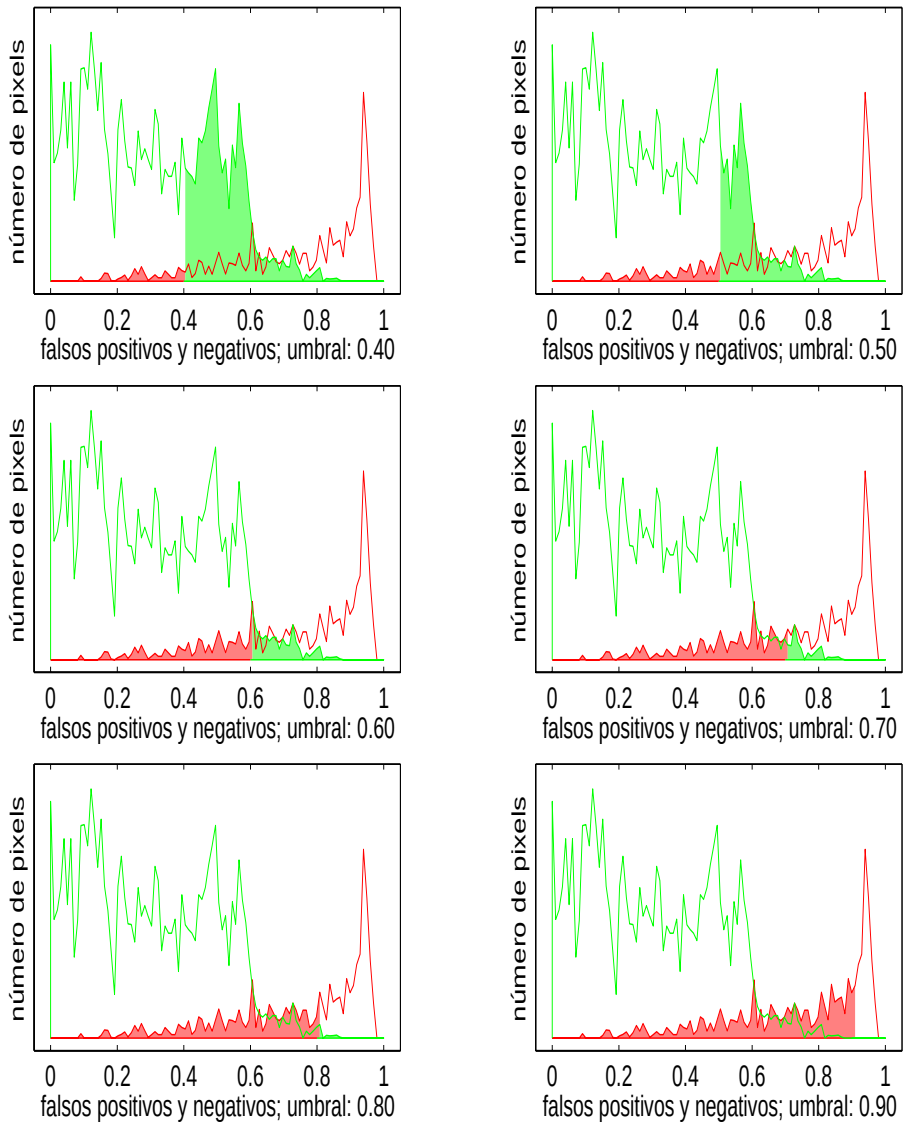


Figura 6.44: Falsos positivos y falsos negativos en función del umbral.

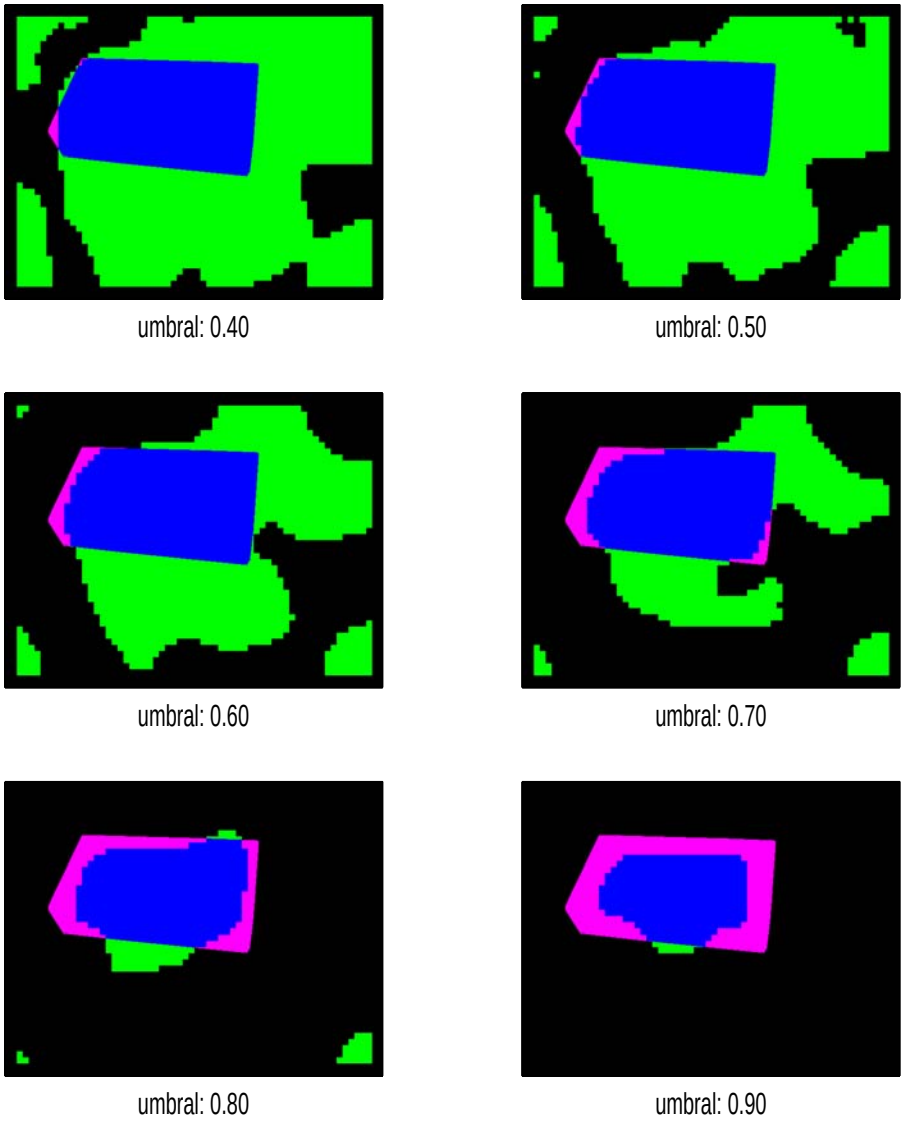


Figura 6.45: Efecto de la variación de los porcentajes de error en función del umbral.

6.4.4. Uso de los umbrales

En la etapa de detección multiescala NMFSC, sólo se obtiene un umbral automático, ν_{ON-OFF}, puesto que sólo se dispone del modelo de apariencia NMFSC en el canal ON-OFF. Este umbral es el que se utiliza junto con el mapa de probabilidad global, resultado de la ponderación multiescala, para decidir si el objeto está o no presente en la escena y en caso positivo, realizar su segmentación.

Por otro lado, en la etapa de detección multicanal PCA se obtiene un umbral, ν_i, para cada uno de los diferentes canales i. Sin embargo, no se utilizarán directamente sobre cada uno de los mapas de probabilidad de cada canal sino que se combinarán para obtener un umbral global, ν_G, que sea el responsable de la segmentación final del objeto mediante su uso sobre el mapa de probabilidad global. Este umbral se obtendrá a través de una expresión equivalente a la utilizada para el cálculo del mapa de probabilidad global:

$$\nu_G = \frac{K_1 \cdot \nu_1 + K_2 \cdot \nu_2 + \ldots + K_N \cdot \nu_N}{K_1 + K_2 + \ldots + K_N} \tag{6.26}$$

donde los coeficientes K_i son los pesos de los canales obtenidos mediante el procedimiento descrito en el apartado 6.3.

El primer método (basado en las distribuciones espaciales de los modelos) proporciona un umbral que es una distancia, luego antes de utilizar la expresión 6.26 para el cálculo de ν_G se debe normalizar el umbral de la misma forma que se normalizaban los mapas de distancia $\mathcal{D}_i$ en mapas de probabilidad $\mathbf{M}_i$ (apartado 6.1.1.2).

En el segundo método propuesto para la obtención del umbral (mediante aprendizaje con escenas etiquetadas), éste se obtiene como un valor de probabilidad puesto que se trabaja con los mapas de probabilidad normalizados de cada canal, $\mathcal{M}_i$, por lo que la expresión 6.26 es aplicable directamente una vez son calculados los umbrales de cada canal. Sin embargo, puesto que también es posible obtener el mapa de probabilidad global a través de la expresión 6.23, otra forma de obtener este umbral global ν_G es utilizando sólo los mapas de probabilidad globales de las escenas etiquetadas debido a la linealidad de ambas expresiones.

Capítulo 7

Resultados experimentales

En este capítulo se describe el entorno de pruebas utilizado para la aplicación de detección de objetos, presentada en el capítulo anterior. Todas las aplicaciones desarrolladas se han realizado sobre Matlab para facilitar la visualización de todas las etapas del proceso de detección. Las imágenes de los objetos, tanto las de entrenamiento como las de test, pueden capturarse utilizando una sencilla webcam.

Además, se presenta un estudio detallado de la capacidad de detección de los distintos canales que forman parte de la etapa de detección multicanal. También se analiza la capacidad de detección de los canales PCA combinados, así como la eficiencia del canal ON-OFF para la etapa de detección multiescala. Por último, se presenta un estudio de la validez de emplear ambas etapas (PCA y NMFSC) combinadas.

7.1. Entorno de pruebas utilizado

Se ha desarrollado un entorno de pruebas en Matlab capaz de realizar tanto la fase de entrenamiento (captura de imágenes, procesado, generación de los modelos, ajuste de los pesos de los canales y obtención del umbral) como la fase de funcionamiento automático del detector.

7.1.1. Captura de imágenes con webcam

Tanto las imágenes de entrenamiento de los objetos como las imágenes de test pueden ser capturadas desde una sencilla webcam, puesto que nuestro detector no requiere utilizar imágenes de alta resolución. La figura 7.1 ilustra la captura de imágenes de entrenamiento del objeto de interés mediante una webcam.

Figura 7.1: Captura de imágenes de entrenamiento con una webcam.

Si se desea que el detector reconozca al objeto en cualquier pose, es necesario que las imágenes de entrenamiento cubran ampliamente todas las vistas diferentes de objeto.

A partir de un conjunto de imágenes de entrenamiento, como el mostrado en la figura 7.2 para el objeto *caja amarilla*, se realizan los procesos *off-line*, es decir, el entrenamiento del sistema de detección. Cuanto más representativo de la apariencia del objeto sea el conjunto de imágenes de entrenamiento, mejor será el modelo de apariencia obtenido, luego es necesario que el conjunto de imágenes de entrenamiento cubra toda la variabilidad de las vistas del objeto.

Las imágenes de entrenamiento completas sirven de entrada a la generación del modelo NMFSC utilizado en la etapa de detección multiescala. Pero para construir los modelos PCA se emplean fragmentos cuadrados de las imágenes de entrenamiento.

Figura 7.2: Conjunto de imágenes de entrenamiento.

7.1.2. Fragmentos de las imágenes

Los fragmentos de las imágenes se obtienen de forma aleatoria a partir del conjunto de entrenamiento. Para aprovechar las ventajas de utilizar fragmentos, resulta interesante que el conjunto de entrenamiento recopile imágenes del objeto a diferentes distancias respecto de la cámara, para conseguir una mayor variabilidad en los fragmentos y mejorar la capacidad del sistema de detección de operar aún cuando el objeto no se encuentra a una distancia fija de la cámara. La figura 7.3 muestra un ejemplo de un conjunto de fragmentos de la *caja amarilla*, y la figura 7.4 ilustra el proceso de obtención de los fragmentos en los distintos espacios de color utilizados por el sistema.

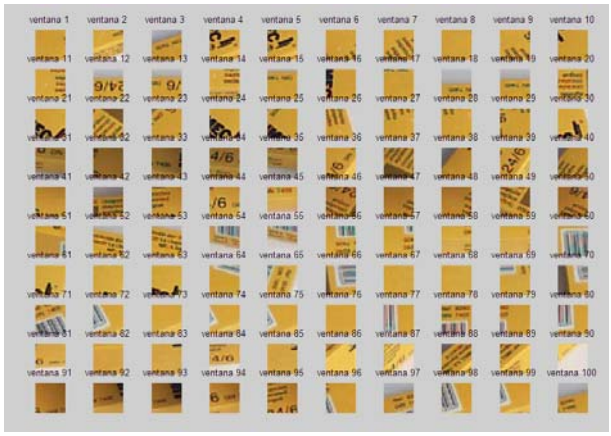


Figura 7.3: Conjunto de fragmentos de las imágenes de entrenamiento de la *caja amarilla*.

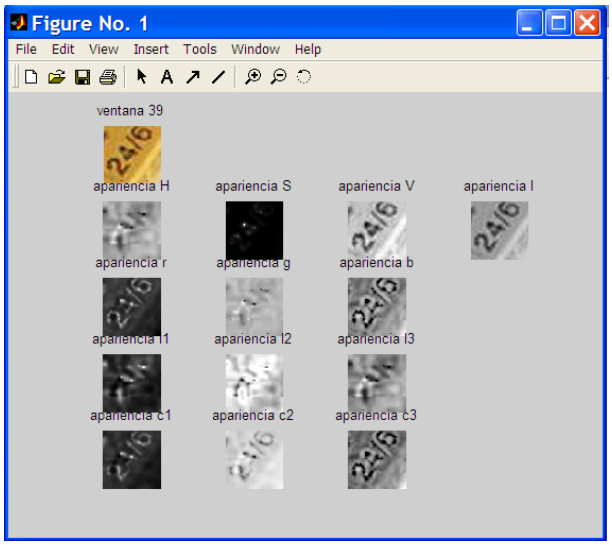


Figura 7.4: Obtención de los fragmentos de las imágenes de entrenamiento de la *caja amarilla* en los distintos espacios de color.

7.1.3. Base de datos de objetos a reconocer

Para las pruebas se han utilizado múltiples objetos de apariencia muy diversa. La figura 7.5 muestra algunos de ellos:

En cualquier caso, la gran variabilidad existente en condiciones de iluminación, distancia a los objetos, orientación de los mismos, fondos, etc., hace que el número de objetos no ofrezca una idea adecuada de la inmensa variedad de situaciones que es posible alcanzar utilizando la librería de objetos propuesta. Por tanto, no se considera necesario aumentar el número de objetos para probar la capacidad de reconocimiento del sistema.

7.1.4. Condiciones de captura de las escenas para la detección

El escenario mostrado en la figura 7.1 permite ver que se ha decidido no controlar excesivamente las condiciones de captura, para hacer los experimentos lo más similares posibles a una aplicación real. Hay diversos aspectos de las condiciones de captura en los que se desea hacer énfasis:

- ***Iluminación***

  La iluminación no está controlada: algunas de las imágenes han sido capturadas con luz natural, a diversas horas del día; mientras que otras han sido capturadas con distintos tipos de iluminación artificial (varios tipos de focos, no siempre situados en la misma orientación). El resultado son imágenes con aspectos muy distintos, con algunos brillos y con sombras en diversas orientaciones, tal como muestran las figuras 7.6 y 7.7.

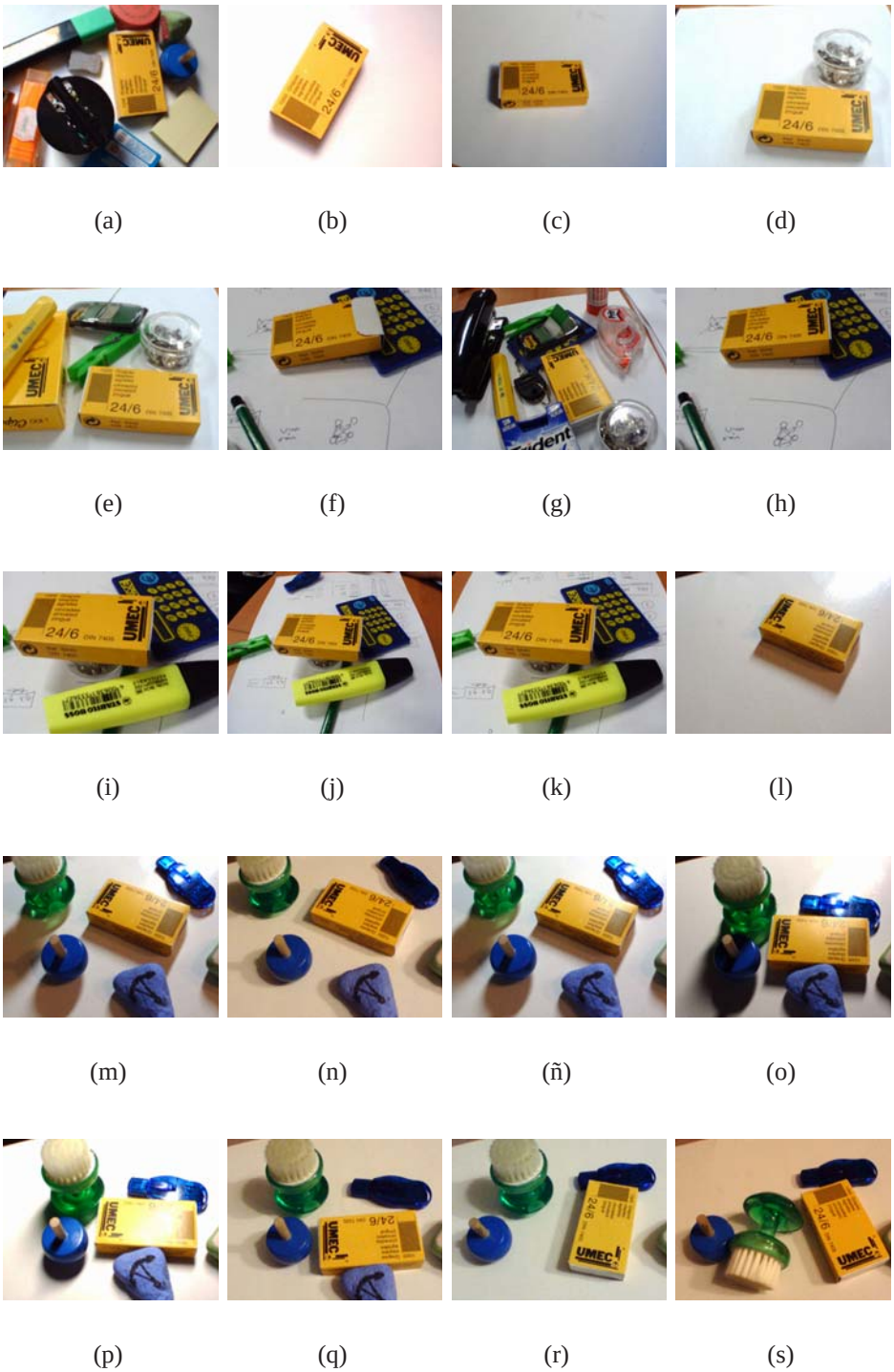
Figura 7.5: Algunos de los objetos utilizados durante las pruebas.

- ***Fondos***

  Los fondos también presentan una gran variabilidad. En algunas de las escenas, el fondo es uniforme, pero en general, los fondos son no uniformes, bien por el tipo de superficie sobre la que se sitúa el objeto (por ejemplo, madera), como por la presencia de otros objetos circundantes, que incluso pueden ocluir parcialmente al objeto de interés. Las figuras 7.6 y 7.7 muestran algunas escenas utilizadas con fondos muy diversos.

- ***Distancia al objeto***

  La distancia al objeto de interés tampoco se mantiene fija: en primer lugar, la situación de la cámara no siempre es la misma (ni su orientación con respecto a la superficie horizontal donde se colocan los objetos); y en segundo lugar, los objetos se sitúan aleatoriamente. Con lo que no siempre se hayan a la misma distancia de la cámara. La variedad de orientaciones en las que pueden situarse los objetos hacen que la distancia a la cámara también varíe, sobre todo si las dimensiones ancho $\times$ alto $\times$ largo no son homogéneas (como es el caso de la caja amarilla). Las figuras 7.6 y 7.7 muestran algunas escenas donde que patente la no uniformidad de las distancias.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (ñ) (o)

(p) (q) (r) (s)

Figura 7.6: Ejemplos de escenas utilizadas para el entrenamiento y durante la detección.

(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (ñ) (o)

(p) (q) (r) (s)

Figura 7.7: Ejemplos de escenas utilizadas para el entrenamiento y durante la detección.

7.1.5. Condiciones de captura de las imágenes para la generación de los modelos de cada objeto

El modelo de cada objeto consta de diversas imágenes del mismo (desde un mínimo de 10 hasta un máximo de 50) en múltiples orientaciones. A diferencia de algunas bases de datos existentes, en las que las imágenes se han capturado rotando los objetos a intervalos regulares (por ejemplo 5 grados en la base de datos COIL-100), para los experimentos de la presente tesis se ha decidido utilizar un procedimiento más parecido al que podría emplear un usuario en una aplicación real del sistema. Las imágenes de entrenamiento se capturan manualmente, rotando bien el objeto o bien la cámara sin necesidad de ningún tipo de mecanismo o plataforma motorizada. Se busca cubrir la máxima información posible del objeto, aunque los intervalos entre una imagen y otra no sean regulares. En la figura 7.8 se muestran un conjunto de imágenes de entrenamiento para uno de los objetos utilizados durante las pruebas (un pequeño joyero de madera). En dicha figura se aprecia que las vistas del objeto cubren toda su apariencia externa e incluso el interior del objeto, puesto que se trata de un objeto que presenta una apariencia distinta según si aparece abierto o cerrado.

Figura 7.8: Un conjunto de imágenes de entrenamiento para uno de los objetos utilizados durante las pruebas.

7.2. Aplicaciones desarrolladas

Todas las aplicaciones desarrolladas se han realizado sobre Matlab para facilitar la visualización de los procesos durante todas las etapas de sistema de detección: procesos *off-line* (generación de los modelos de apariencia, ajuste de los pesos automático, cálculo de los umbrales) y *on-line* (detección del objeto en la escena compleja). Además el hecho de emplear Matlab y trabajar con la información visual de la escena en ficheros aporta una gran flexibilidad al sistema, pues es posible utilizar cualquier tipo de aplicación de captura (para webcams o para otro tipo de cámaras).

Se ha optado por mantener el sistema en el entorno Matlab (en lugar de utilizar un lenguaje de programación que pudiera ofrecer más velocidad en ejecución, como C o C++) para facilitar también posibles continuaciones del proyecto. Se considera que no es el objetivo del presente trabajo desarrollar una aplicación lista para ser utilizada, sino sentar las bases de un sistema de detección de objetos y facilitar que este sistema sea desarrollado en más profundidad por otros investigadores.

En cualquier caso, la traducción a otro lenguaje de programación con mayor velocidad en ejecución sería simple en el caso de decidir realizar una aplicación. También sería simple (de acuerdo con la estructura multicanal y multiescala propuesta) la implementación paralela en diversos procesadores para aumentar aún más la velocidad de la detección.

Básicamente, las aplicaciones desarrolladas se pueden clasificar en tres tipos:

- Entorno de entrenamiento
- Entorno de experimentación
- Entorno de funcionamiento automático

El código de todas las funciones de Matlab que dan lugar a estos tres entornos de trabajo se adjuntan dentro del CD disponible junto a esta tesis. A continuación, se describen cada uno de estos entornos de trabajo.

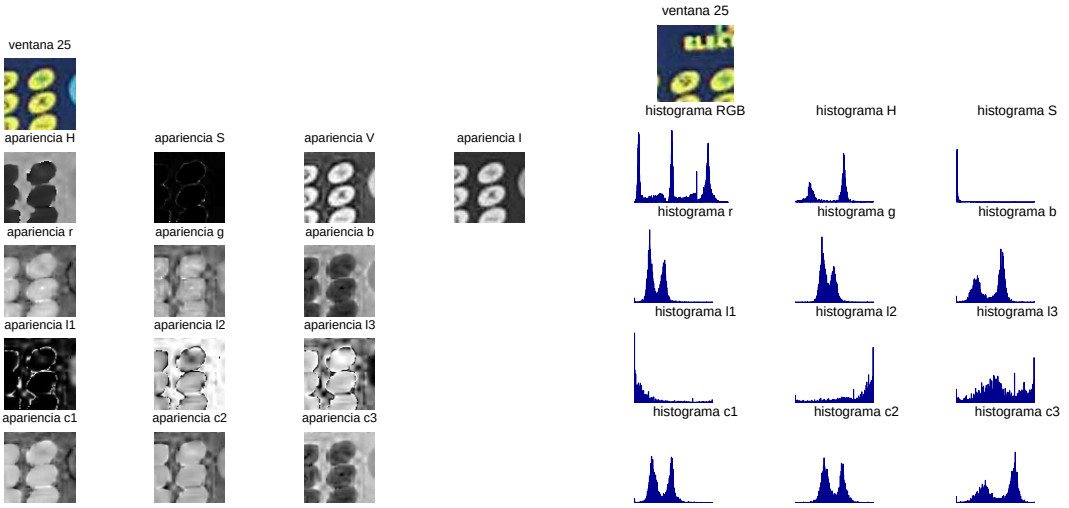
7.2.1. Entorno de entrenamiento

El entorno de entrenamiento permite realizar las siguientes operaciones descritas en el capítulo anterior:

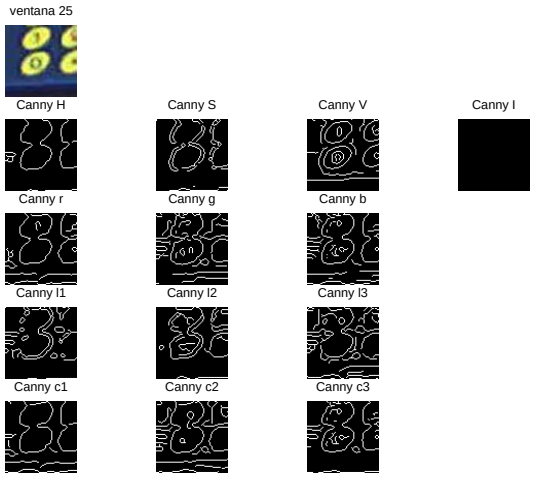
- Obtención de fragmentos.
- Creación de imágenes etiquetadas (escenas complejas etiquetadas).
- Generación de los modelos de apariencia multicanal (PCA).
- Generación del modelo de apariencia (NMFSC) para la etapa de detección multiescala.
- Ajuste de los pesos de los canales automático.
- Cálculo de los umbrales para ambas etapas con los dos métodos propuestos (basado en modelos y basado en escenas etiquetadas).

Las aplicaciones del entorno de entrenamiento necesitan un conjunto de imágenes de entrenamiento del objeto que se pretende detectar (como las de las figuras 7.2 y 7.8) y un conjunto de escenas complejas que cumplan las condiciones del apartado 7.1.4. A partir del conjunto de entrenamiento, una aplicación obtiene automáticamente los fragmentos de las imágenes. El tamaño de estos fragmentos es un parámetro del sistema y puede ser optimizado si se realiza un estudio detallado de los fragmentos del objeto de interés, utilizando el concepto de *merit* visto en el capítulo anterior. La figura 7.9 muestra el proceso de obtención de los fragmentos para uno de los objetos utilizados durante las pruebas.

Para el cálculo de los valores de los pesos y de los umbrales de cada canal se utilizan escenas complejas etiquetadas. Este proceso de etiquetado se realiza manualmente mediante una sencilla aplicación que permite recortar de la escena la zona donde se encuentra el objeto. Mediante el ratón, el usuario selecciona una serie de puntos alrededor de la zona donde se encuentra el objeto en la escena y automáticamente la aplicación generará una forma poligonal cuya área es equivalente al área ocupada por el objeto en dicha escena. A partir de la forma poligonal se construye fácilmente una imagen etiquetada de la escena, es decir, una imagen binaria. Habrá que repetir este proceso para cada una de las escenas que utilicemos para el cálculo de los pesos y los umbrales. La figura 7.10 ilustra este proceso de creación de etiquetas.

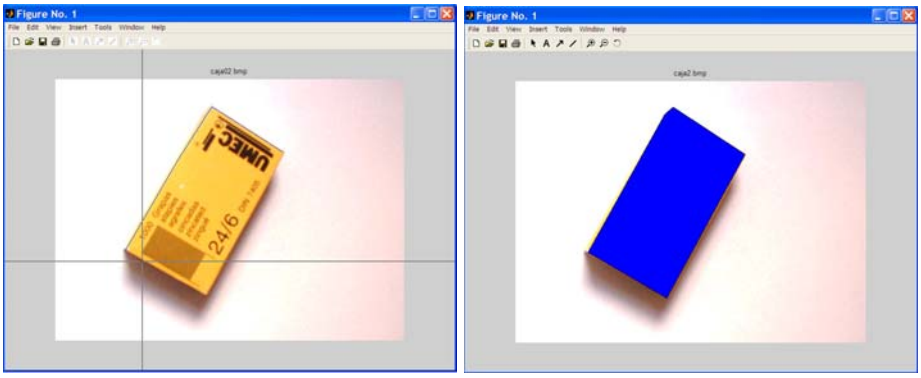


(a) Apariencia (b) Histogramas



(c) Bordes

Figura 7.9: Proceso de obtención de los fragmentos.



(a) Selección de puntos (b) Forma poligonal

Figura 7.10: Creación de escenas etiquetadas.

La generación de los modelos de apariencia multicanal (PCA) y del modelo de apariencia (NMFSC) para la etapa de detección multiescala se realiza de forma automática a partir de las imágenes de entrenamiento o sus fragmentos (en el caso de PCA). Por tanto, los parámetros de entrada a considerar relativos a este programa son: el número de fragmentos (o imágenes) a utilizar ($N_{ejemplos}$) y el número de vectores del subespacio ($N_{vectores}$), es decir, la dimensión de las proyecciones de las imágenes de entrenamiento. Cuanto mayores sean ambos parámetros, el modelo de apariencia del objeto será más robusto, pues contendrá mayor información del objeto. Sin embargo, una dimensión alta y un número elevado de puntos en el modelo produce que el clasificador utilizado se ralentice. Por tanto, el usuario debe decidir, en función de la aplicación final, los valores a utilizar de estos dos parámetros. En las pruebas realizadas durante el estudio de la capacidad de detección del sistema propuesto, los valores utilizados para estos dos parámetros son los siguientes:$N_{ejemplos} \in [100, 200]$ y $N_{vectores} \in [5, 50]$

Como ya se detalló en el capítulo anterior, el ajuste de los pesos de los canales se realiza automáticamente a partir de un conjunto de escenas de entrenamiento etiquetadas junto con los mapas de probabilidad generados por el sistema para cada canal. Este proceso se realiza dentro del entorno de entrenamiento informando simplemente del número de escenas que se emplearán para el ajuste.

De forma similar, el cálculo de los umbrales, por las dos metodologías se realiza a partir de información ya presente en el sistema: los modelos de apariencia generados y las escenas etiquetadas.

7.2.2. Entorno de experimentación para comprobar la influencia de cada canal y el efecto de los umbrales

Con el objetivo de comprobar el funcionamiento adecuado de cada canal y el efecto de aplicar los umbrales sobre los mapas de probabilidad de la escena, se han desarrollado dos aplicaciones: la primera de ellas, denominada *visualiza mapas*, permite la visualización de todos los mapas de probabilidad, con la posibilidad de realzar o no los canales según el factor de escalado configurable γ (descrito en ecuación 6.25); y la otra aplicación, *visualizador interactivo*, permite tanto la visualización de una selección de mapas de probabilidad, como el ajuste manual de los pesos y umbrales.

A continuación, se muestran ejemplos de visualización de mapas de probabilidad, tanto relativos a la búsqueda multicanal como a la búsqueda multiescala, para ambas aplicaciones.

La figura 7.12 muestra los 38 mapas de probabilidad de la etapa de búsqueda multicanal PCA obtenidos para la escena de la figura 7.11 utilizando la aplicación *visualiza mapas*. El último gráfico (*PROMEDIO*) de la figura 7.12 muestra el resultado de promediar todos los canales con pesos de igual valor. La figura 7.13 muestra los mapas de probabilidad anteriores pero realzados, es decir, se ha realizado un escalado no lineal de valor $\gamma = 5$ en cada uno de ellos.

Los mapas de probabilidad de la etapa de búsqueda multiescala NMFSC de la escena de la figura 7.11 se muestran en la figura 7.14, donde cada mapa de probabilidad equivale a una búsqueda del objeto en una escala diferente: en este caso, los factores de búsqueda multiescala utilizados en la escena son: $0{,}6, 0{,}8, 1{,}0, 1{,}2, 1{,}4, 1{,}6, 1{,}8, 2{,}0, 2{,}2, 2{,}4, 2{,}6$ y $2{,}8$. La figura 7.15 muestra otro ejemplo de mapas de probabilidad obtenidos tras la búsqueda multiescala para otra escena compleja.

Figura 7.11: Escena para la detección del objeto *la caja amarilla* .

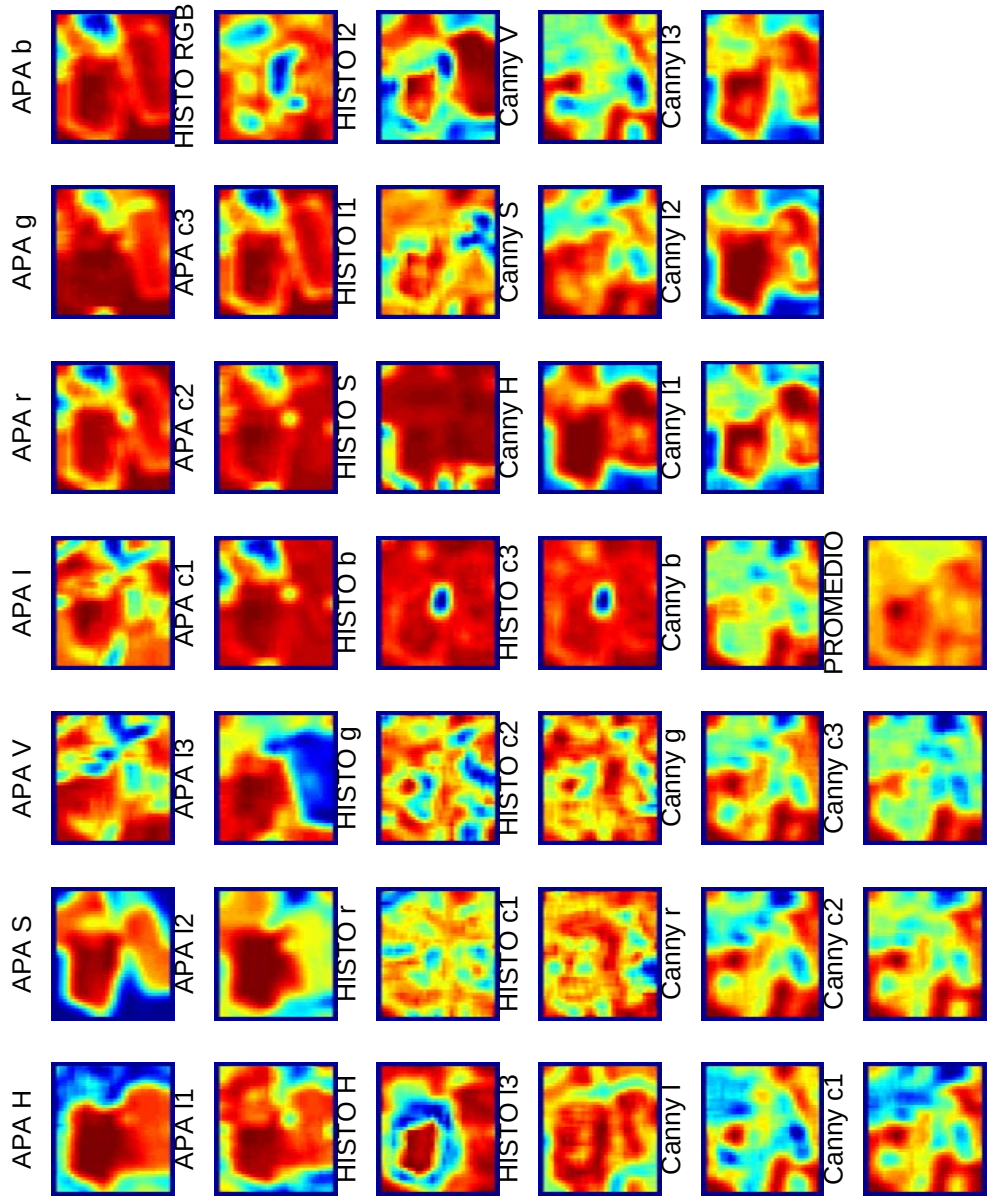


Figura 7.12: Entorno *visualiza mapas*: los 38 mapas de probabilidad de la etapa de búsqueda multicanal PCA obtenidos para la escena de la figura 7.11.

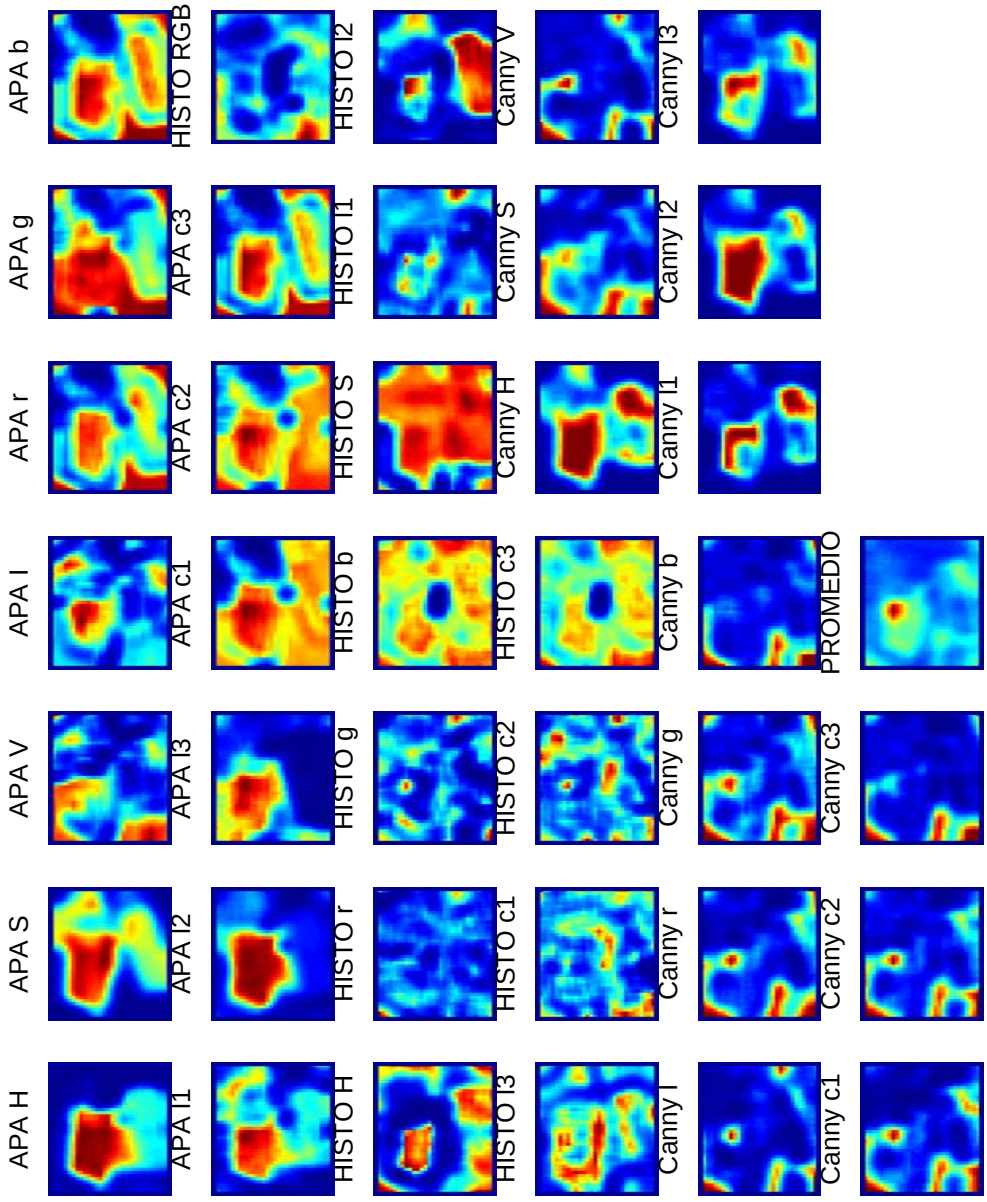


Figura 7.13: Entorno *visualiza mapas*: los 38 mapas de probabilidad de la etapa de búsqueda multicanal PCA obtenidos para la escena de la figura 7.11 con realce $\gamma = 5$.

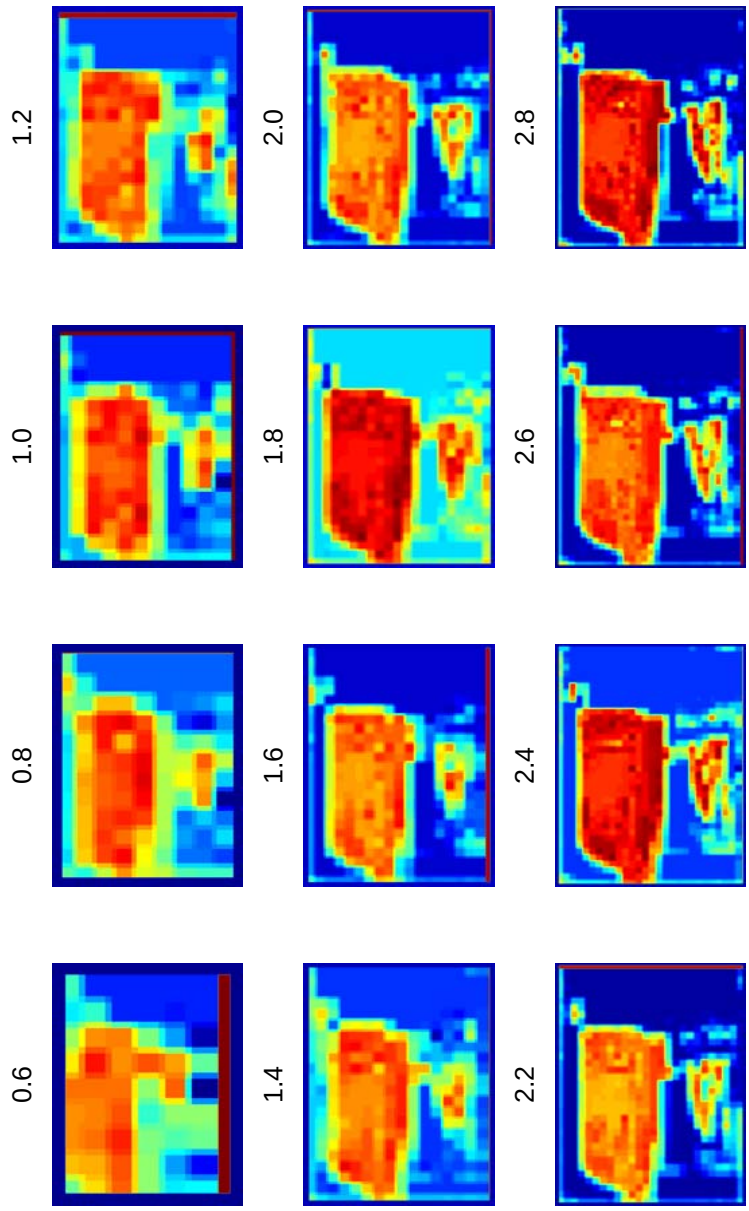


Figura 7.14: Entorno *visualiza mapas*: mapas de probabilidad de la etapa de búsqueda multiescala NMFSC de la escena de la figura7.11 (Los factores de escalado de la escena vienen expresados en tanto por uno).

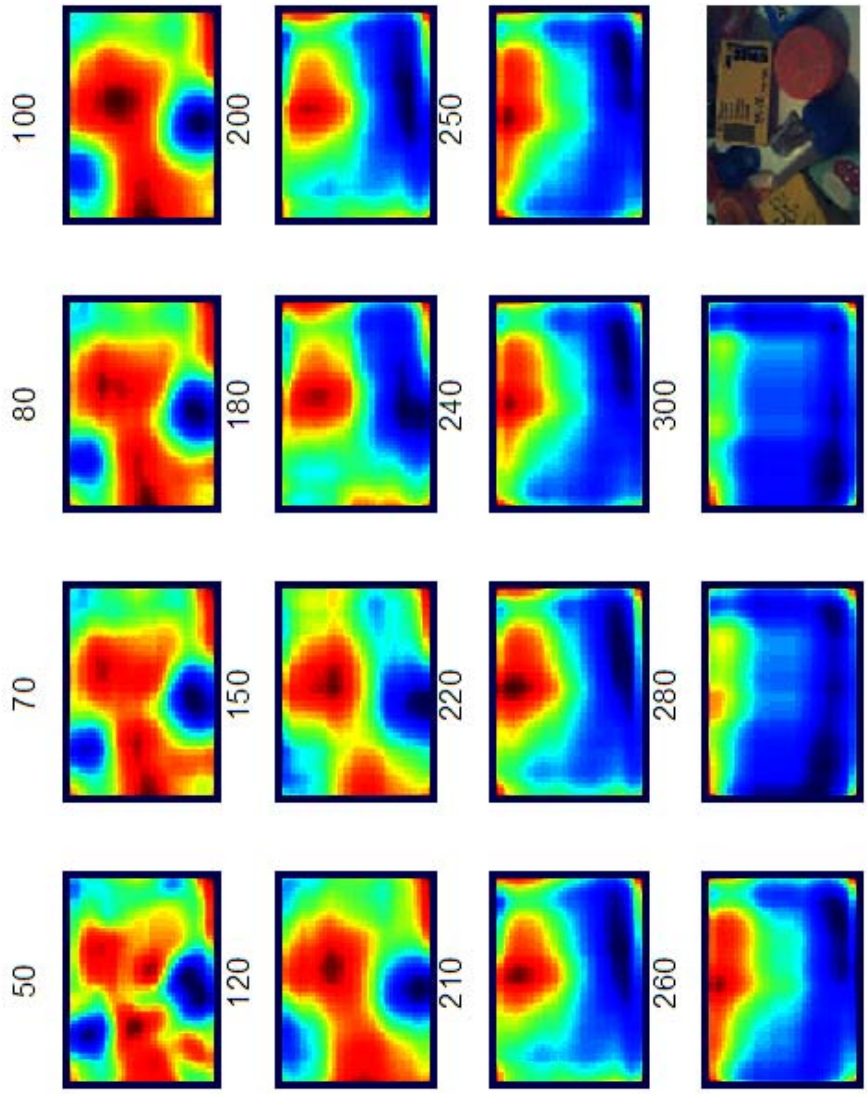


Figura 7.15: Entorno *visualiza mapas*: mapas de probabilidad de la etapa de búsqueda multiescala NMFSC de otra escena compleja (se muestra en la esquina inferior derecha). Se aprecia como ciertas escalas producen resultados muy satisfactorios, mientras que en otras escalas el objeto desaparece completamente. (Los factores de escalado de la escena en este caso vienen expresados en tanto por cien).

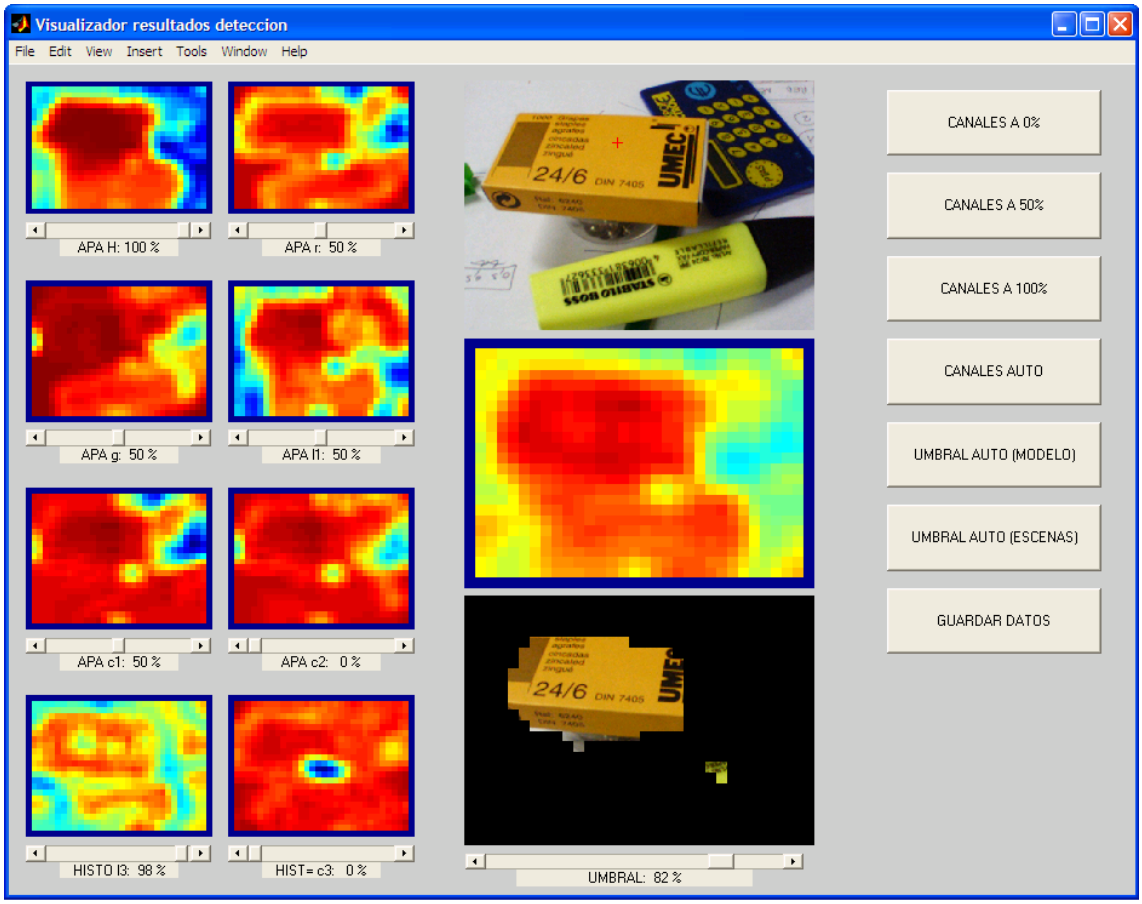


Figura 7.16: Entorno *visualizador interactivo*: permite visualizar los mapas de probabilidad obtenidos durante las etapa de búsqueda y examinar el efecto de los pesos y el umbral aplicado.

El aspecto del *visualizador interactivo* se muestra en la figura 7.16. En la parte izquiera del mismo, se muestran los mapas de probabilidad de diversos canales (se puede visualizar un máximo de 8 canales). En la parte central, se muestran, de arriba a abajo, la escena original, la suma ponderada de todos los mapas de probabilidad, y el conjunto de píxeles que superan un cierto umbral (píxeles donde es esperable que se encuentre el objeto de interés). Adicionalmente, sobre la escena original se marca una cruz en el punto de máxima probabilidad, de acuerdo con la suma ponderada de todos los mapas.

La parte derecha del visualizador contiene diversos botones que, junto con las barras de desplazamiento o cursores situadas debajo de cada mapa de probabilidad y debajo de la escena umbralizada, permiten un ajuste manual del detector para la experimentación. Las barras de desplazamiento situadas bajo los mapas de probabilidad permiten ponderar por

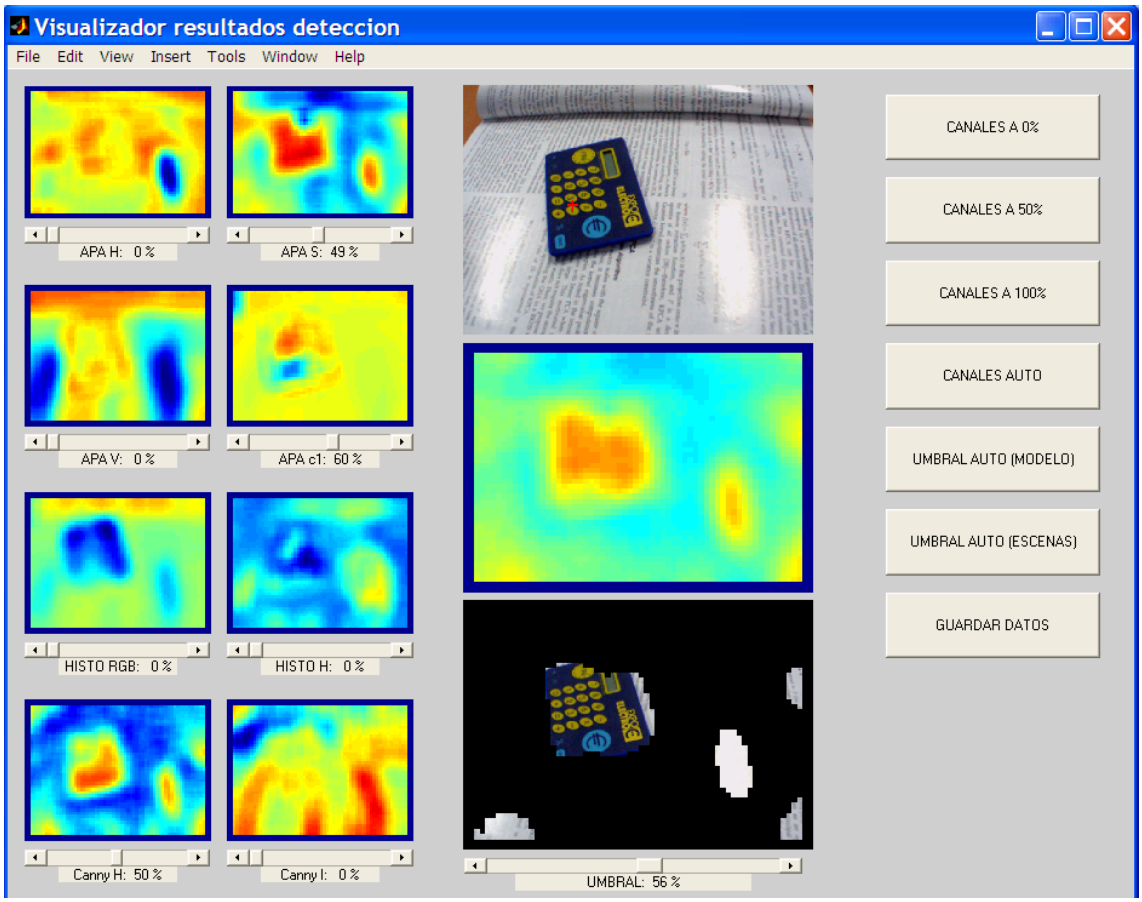


Figura 7.17: Entorno *visualizador interactivo*. En este caso, el mapa de probabilidad global se ha obtenido sólo a partir de la ponderación de tres canales (Apariencia S y c_1, y Canny H.) .

separado la influencia de cada canal con respecto de los demás. Para una mayor facilidad en el ajuste de estas influencias o peso, se incluyen tres botones en la parte derecha del visualizador:

- *canales a* $0\,\%$: Permite ajustar el peso de todos los canales al mínimo valor, para realizar rápidamente pruebas en las que sólo uno de los canales influye en el mapa de probabilidad final (sólo hay que mover uno de los cursores).

- *canales a* $100\,\%$: Permite, del mismo modo, ajustar el peso de los canales al máximo valor, para realizar rápidamente experimentos en los que sólo uno de los canales es descartado respecto de los demás.

- *canales a* $50\,\%$: Permite combinar los dos efectos anteriores, de modo que la mayor parte de los canales tengan pesos similares, pero sea fácil anular alguno de ellos y

potenciar algún otro respecto del resto.

- *auto*: Permite restablecer el vector de pesos devuelto por la etapa de entrenamiento con escenas etiquetadas (y pasado como parámetro a la función *visualizador interactivo*).

En cuanto al control del umbral, se dispone de dos botones que permiten situar el umbral automáticamente:

- *umbral auto (modelo)*: Sitúa el umbral en el valor calculado de acuerdo con el modelo de apariencia del objeto.
- *umbral auto (escenas)*: Sitúa el umbral en el valor calculado usando escenas etiquetadas.

En ambos casos, los valores de los umbrales son pasados como parámetro a la función *visualizador interactivo*.

Por último, el umbral también puede ser fijado manualmente mediante la barra de desplazamiento situada bajo la imagen umbralizada, entre un $0\,\%$ (equivalente a probabilidad 0, todos los píxeles se considerarían pertenecientes al objeto) y un $100\,\%$ (equivalente a probabilidad 1, casi ningún píxel se consideraría perteneciente al objeto).

El último botón existente, *guardar*, permite almacenar el resultado completo del experimento: la escena original, los mapas de probabilidad, los pesos y el umbral.

La figura 7.17 muestra otro ejemplo de funcionamiento del *visualizador interactivo*, en este caso, el objeto a detectar es una *calculadora* y en la aplicación sólo se han ponderado tres canales (Apariencia S y c_1, y Canny H.). Las figuras 7.18 y 7.19 muestran el *visualizador interactivo* con la misma escena de entrada pero con mapas de probabilidad relativos a búsquedas diferentes: en la figura 7.18, el objeto de interés es la *calculadora*, mientras que la figura 7.19, el objeto a buscar es la *caja amarilla*.

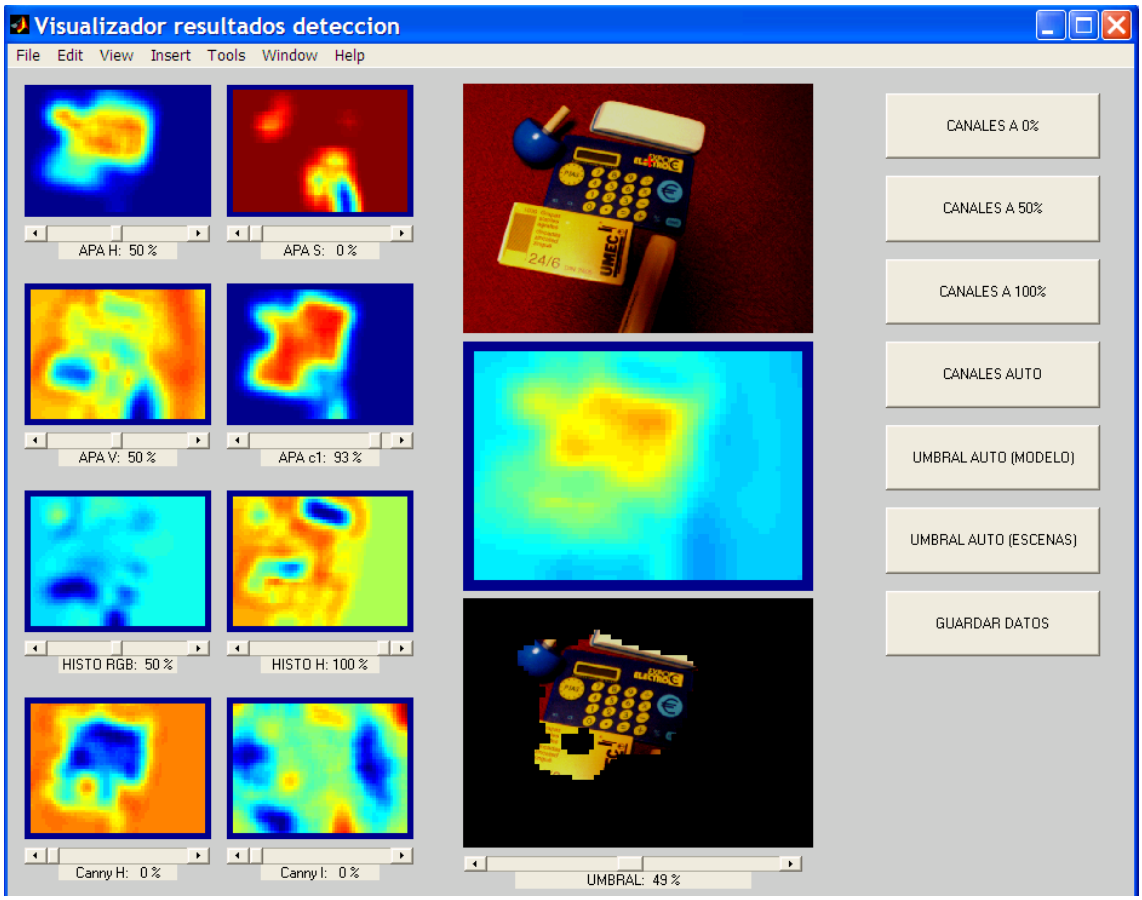


Figura 7.18: Entorno *visualizador interactivo*, objeto a detectar: la *calculadora*.

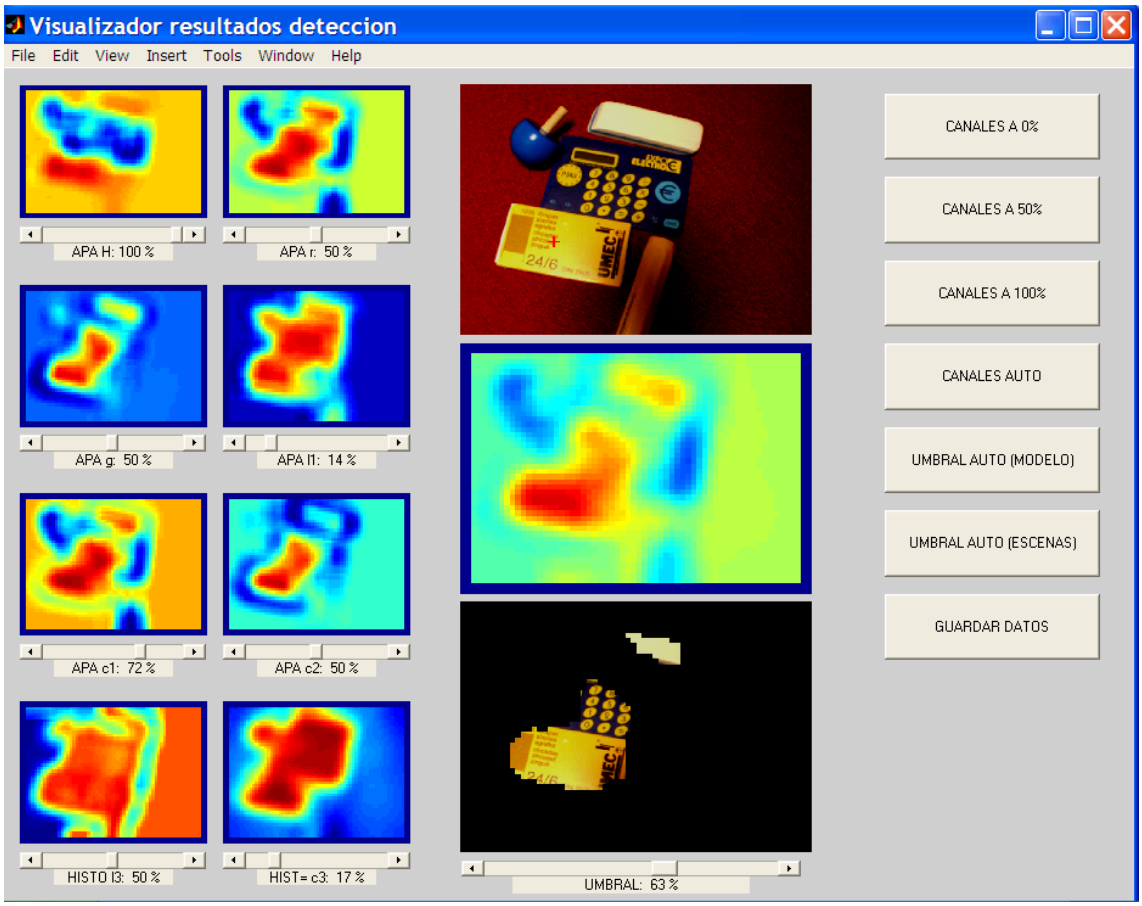


Figura 7.19: Entorno *visualizador interactivo*, objeto a detectar: la *caja amarilla*.

7.2.3. Entorno de funcionamiento automático

El entorno de funcionamiento automático permite realizar la detección del objeto de interés, sobre una imagen de entrada de la escena, a partir de la información generada y almacenada en el sistema durante el entrenamiento (modelos de apariencia, pesos de los canales, K_i y valores de los umbrales globales ν_G).

El entorno de funcionamiento automático dispone de las siguientes aplicaciones:

- etapa de búsqueda multicanal PCA
- etapa de búsqueda multiescala NMFSC
- combinación de etapas
- visualización de resultados

Utilizando la escena a analizar como entrada a la etapa de búsqueda multicanal PCA, ésta devuelve un mapa de probabilidad global que señala las zonas de mayor probabilidad en las que puede hallarse el objeto, a partir de los modelos de apariencia generados durante el entrenamiento y el ajuste de pesos de canales realizado (ecuación 6.4).

De forma equivalente, la etapa de búsqueda multiescala NMFSC proporciona un mapa de probabilidad global relativo al proceso de búsqueda multiescala realizado con el método NMFSC.

Posteriormente, mediante el uso de los umbrales de cada etapa (generados también durante la fase *off-line*), se obtienen sendas escenas umbralizadas en las que se visualiza el área correspodiente al objeto. La intersección de ambas escenas umbralizadas es el resultado final de nuestro sistema de detección.

Los resultados del entorno de funcionamiento automático se visualizan mediante la aplicación *visualizador detección* que proporciona los mapas de probabilidad globales de ambas etapas, las escenas umbralizadas y el resultado de la intersección de éstas. La figura 7.20 muestra el aspecto del *visualizador detección*.

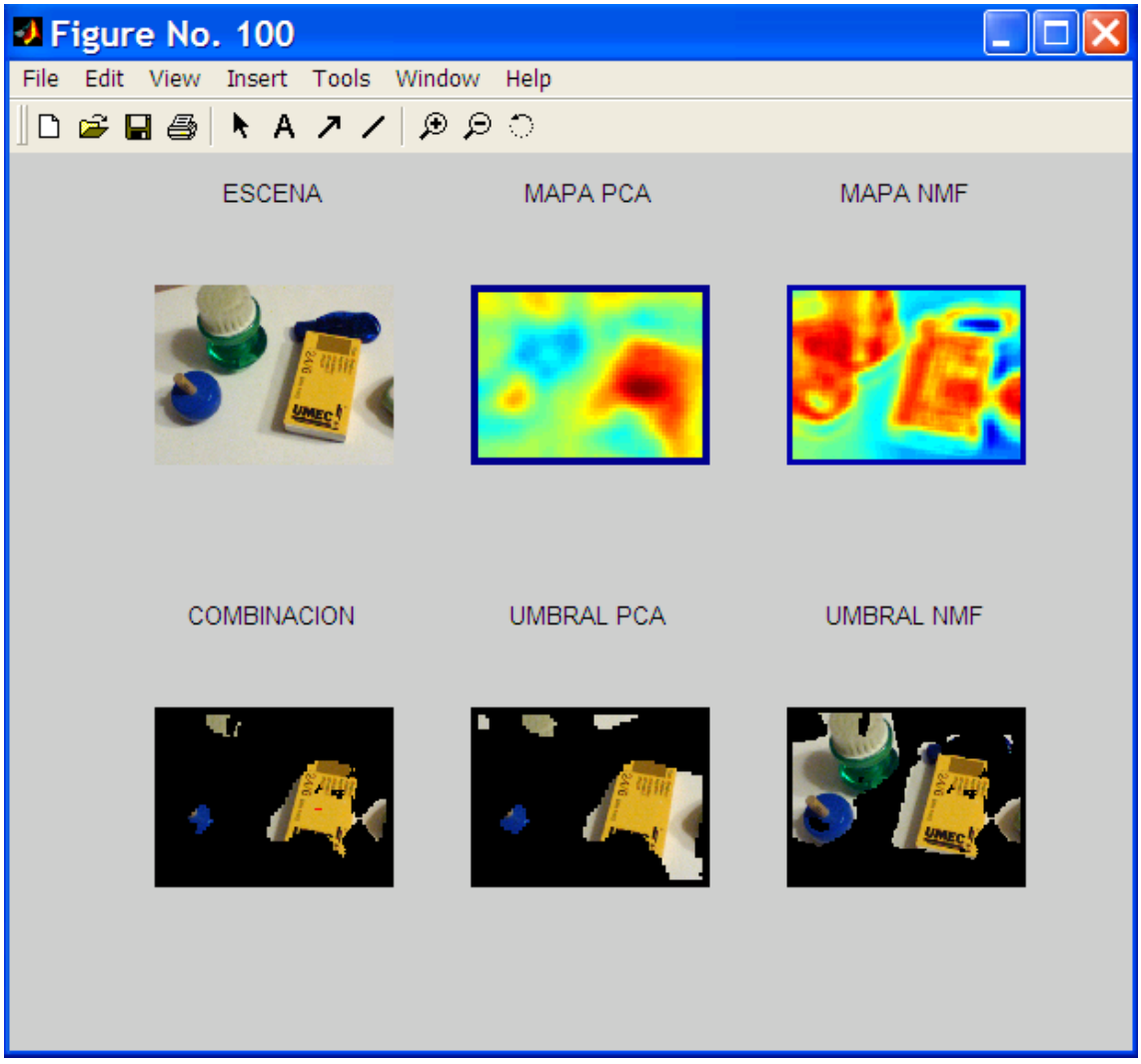


Figura 7.20: Entorno *visualizador deteccion*, objeto a detectar: la *caja amarilla*.La combinación de las escenas umbralizadas de ambas etapas disminuye el número de falsos positivos de cada una de las dos etapas por separado.

7.3. Capacidad de detección de los distintos canales.

En la sección 6.2 se indicó cuáles son los canales que se han utilizado para la detección multicanal PCA. También en la sección 6.3 se indicó cómo en función de los objetos y de las condiciones de iluminacion, fondo, etc., unos canales ofrecen mejores resultados que otros. A continuación, se muestran los resultados obtenidos por cada canal y para cada objeto, en un conjunto de escenas que cubren la mayor variabilidad posible (exactamente, para cada experimento se utilizan 20 escenas con variaciones de iluminación, fondos, proximidad de los objetos a la cámara, orientación de la cámara, etc.).

Los resultados se indican en porcentaje de píxeles clasificados correctamente (como pertenecientes o no pertenecientes al objeto buscado).

Figura 7.21: Objetos utilizados para mostrar la capacidad de detección de los distintos canales.

Todos los valores experimentales se han obtenido mediante un procedimiento de validación cruzada de 5 divisiones (5-*fold cross-validation*) repetido 10 veces para asegurar la fiabilidad de los resultados:

- De las 20 escenas etiquetadas disponibles para cada objeto, cuatro quintas partes (16 escenas) se utilizan para el cálculo automático de la influencia de cada canal (pesos) y para el establecimiento de los umbrales (tanto canal por canal como en conjunto para la media ponderada de todos los canales). Estas escenas constituyen los ejemplos de entrenamiento.
- Las 4 escenas restantes se utilizan como ejemplos de prueba: sobre ellas se aplican los pesos y umbrales calculados anteriormente y se obtienen las zonas en las que

presumiblemente se deben encontrar los objetos de interés. Estas zonas se comparan con las etiquetas y de este modo se obtiene el número de píxeles clasificados como falsos positivos y como falsos negativos.

- El proceso completo incluye 50 experimentos. En primer lugar, cada prueba de validación cruzada requiere realizar cinco experimentos distintos: las 20 escenas de test se dividen en cinco grupos y uno de ellos es separado como conjunto de test en cada experimento. En segundo lugar, estas pruebas se repiten 10 veces, en cada ocasión con un reparto distinto (aleatorio) del conjunto de escenas en cinco grupos. Los resultados ofrecidos son los valores medios y las desviaciones típicas de los valores obtenidos en cada uno de los 50 experimentos.

A continuación se mostrarán los resultados obtenidos al intentar reconocer cuatro objetos distintos de la base de datos utilizada : la caja amarilla, una calculadora, una peonza y una vela. Estos objetos se muestran en la figura 7.21. La figura 7.22 muestra algunas de las escenas complejas que se han utilizado para medir la capacidad de detección de los canales empleados en la etapa de detección multicanal PCA.

La figura 7.23 muestra los resultados de reconocimiento obtenidos por cada canal para el primer objeto. Pueden apreciarse claras diferencias en el comportamiento de los canales; en particular, los canales que ofrecen menores errores son los canales 1 (APA H), 8 (APA l_1), 9 (APA l_2), 10 (APA l_3) y 34 (Canny l_2); todos ellos con porcentajes de píxeles mal clasificados inferiores al 15 %. La media ponderada de todos los canales ofrece unos resultados considerablemente mejores que cualquiera de los canales por separado, con lo queda demostrado el correcto funcionamiento del ajuste automático de pesos.

La figura 7.24 muestra los mismos resultados para el segundo objeto. Cabe destacar que los canales que mejor se comportan son distintos de los del objeto anterior. En este caso, los mejores canales son el 8 (APA l_1), el 22 (HISTO l_3) y el 35 (Canny l_3). Como en el caso anterior, la media ponderada de todos los canales ofrece mejores resultados que el mejor de los canales por separado.

Una situación similar se presenta en las figuras 7.25 y 7.26. De nuevo, puede apreciarse cómo los canales más adecuados para el reconocimiento son distintos en función del objeto, y cómo la media ponderada funciona adecuadamente, en estos dos últimos casos con porcentajes de píxeles clasificados incorrectamente inferiores al $5\,\%$.

Los gráficos de la figura 7.27 ofrece resultados más espectaculares. En esta figura se muestran las tasas de error medias para los cuatro objetos estudiados. Ninguno de los canales por separado es capaz de conseguir resultados tan buenos como en los casos anteriores, dado que ninguno de ellos funciona de modo óptimo para todos los objetos simultáneamente; de hecho, los resultados de los distintos canales son más uniformes

que en las gráficas anteriores. Sin embargo, el canal media ponderada, al utilizar pesos específicos para cada objeto, consigue unos muy buenos resultados, con un porcentaje de píxeles erróneos inferior al $7\,\%$. Queda comprobada la fiabilidad del método de ajuste de pesos, y también queda comprobada la necesidad de establecer pesos específicos en función de los objetos.

Un resumen de todos los resultados obtenidos se muestra en la figura 7.28, donde se representan los píxeles clasificados incorrectamente utilizando la media ponderada de todos los canales tanto para cada objeto por separado como para la media de todos los objetos. Esta figura servirá para establecer comparaciones con la capacidad de detección del canal ON-OFF multiescala y con la combinación de ambos procedimientos (multicanal y multiescala) en el siguiente apartado.

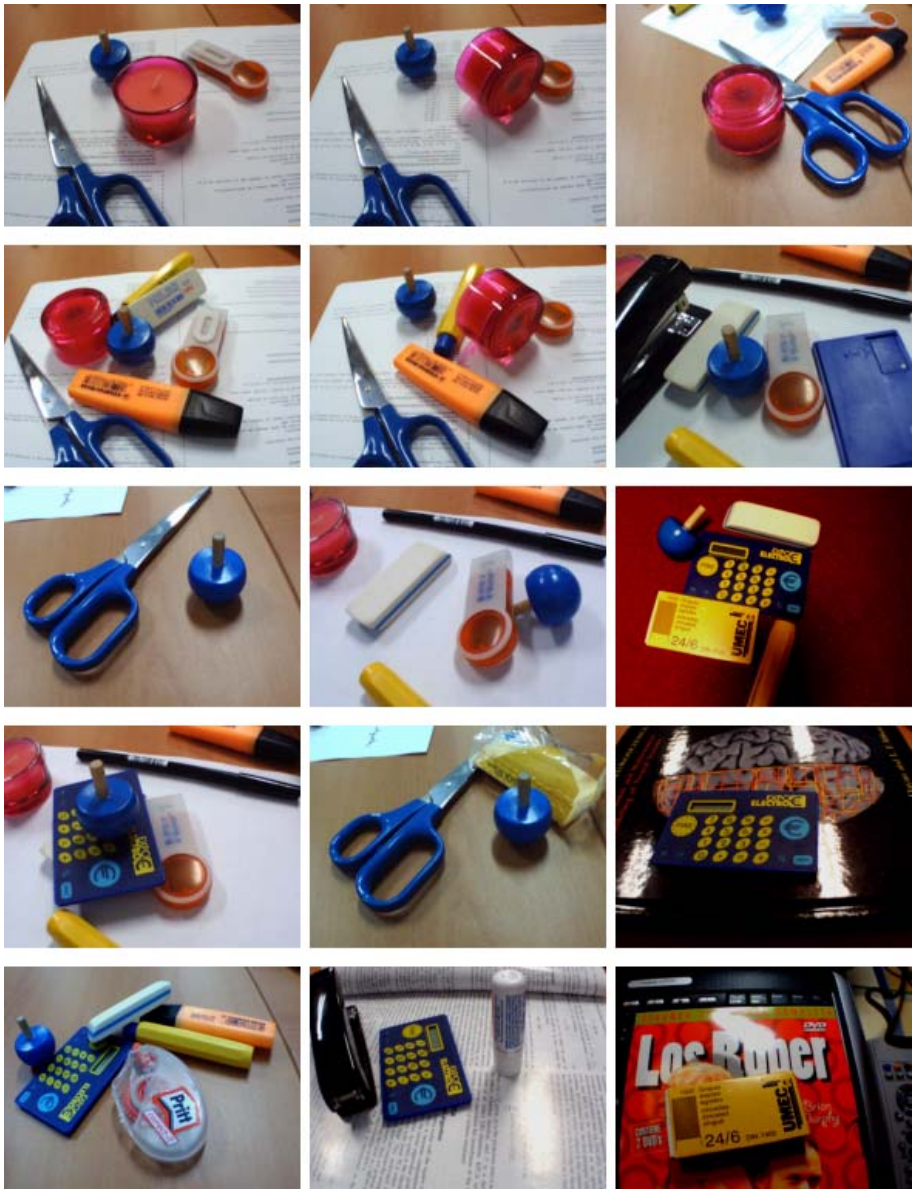


Figura 7.22: Algunas de las escenas complejas utilizadas durante el experimento.

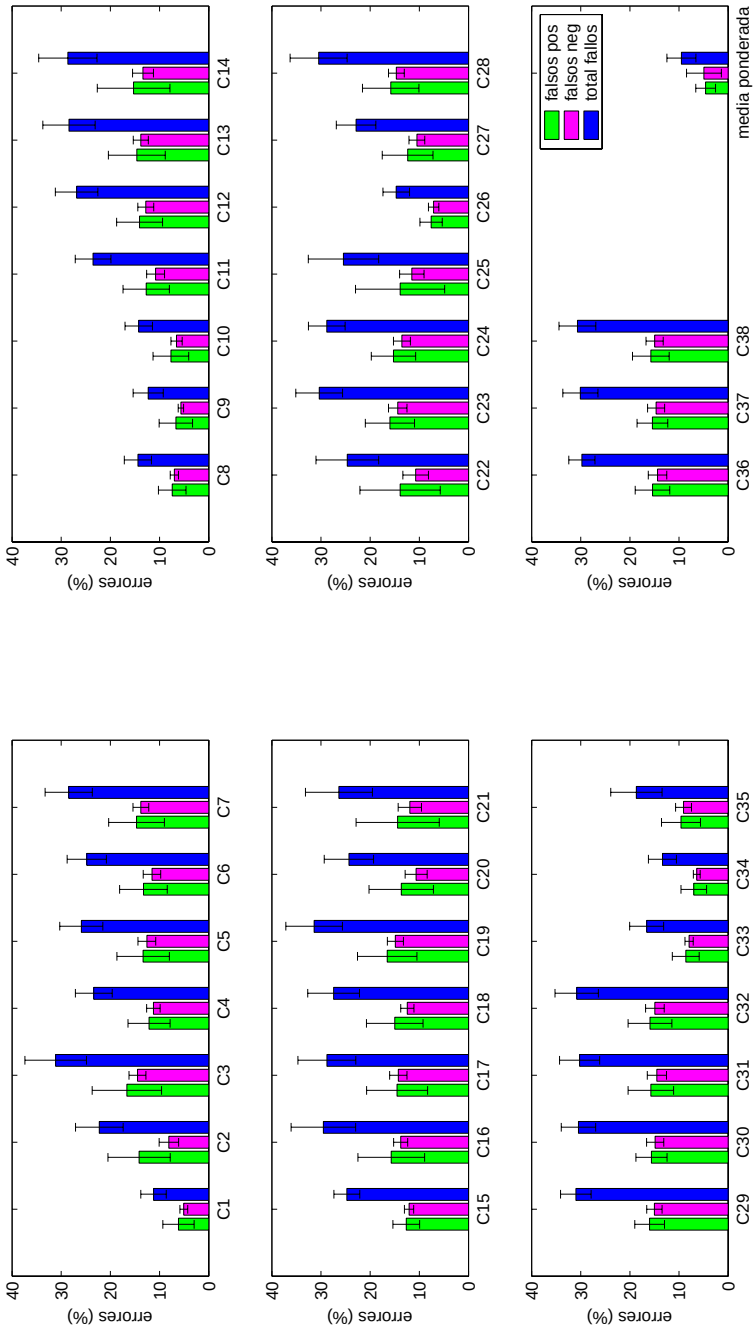


Figura 7.23: Resultados obtenidos para el primer objeto. Se muestran los píxeles clasificados incorrectamente (falsos positivos, falsos negativos y suma de ambos) para todos los canales y para la media ponderada de los mismos.

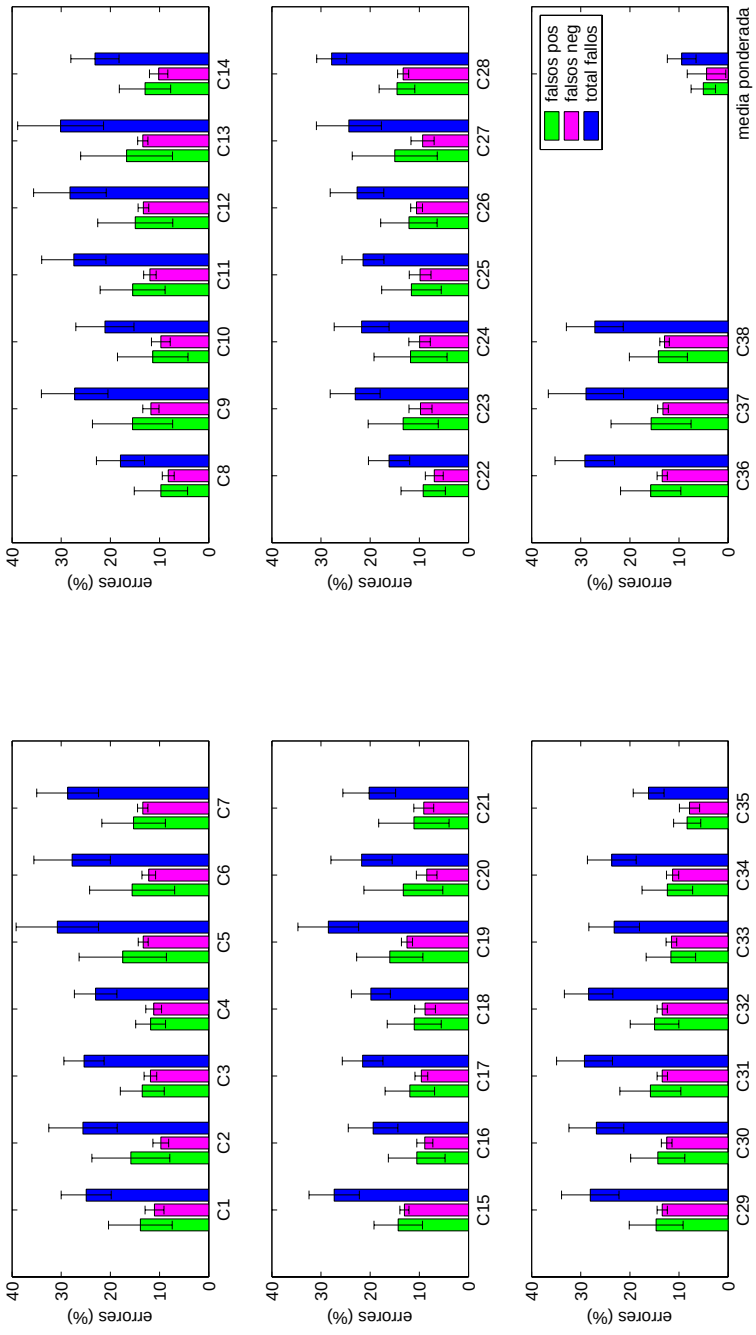


Figura 7.24: Resultados obtenidos para el segundo objeto.

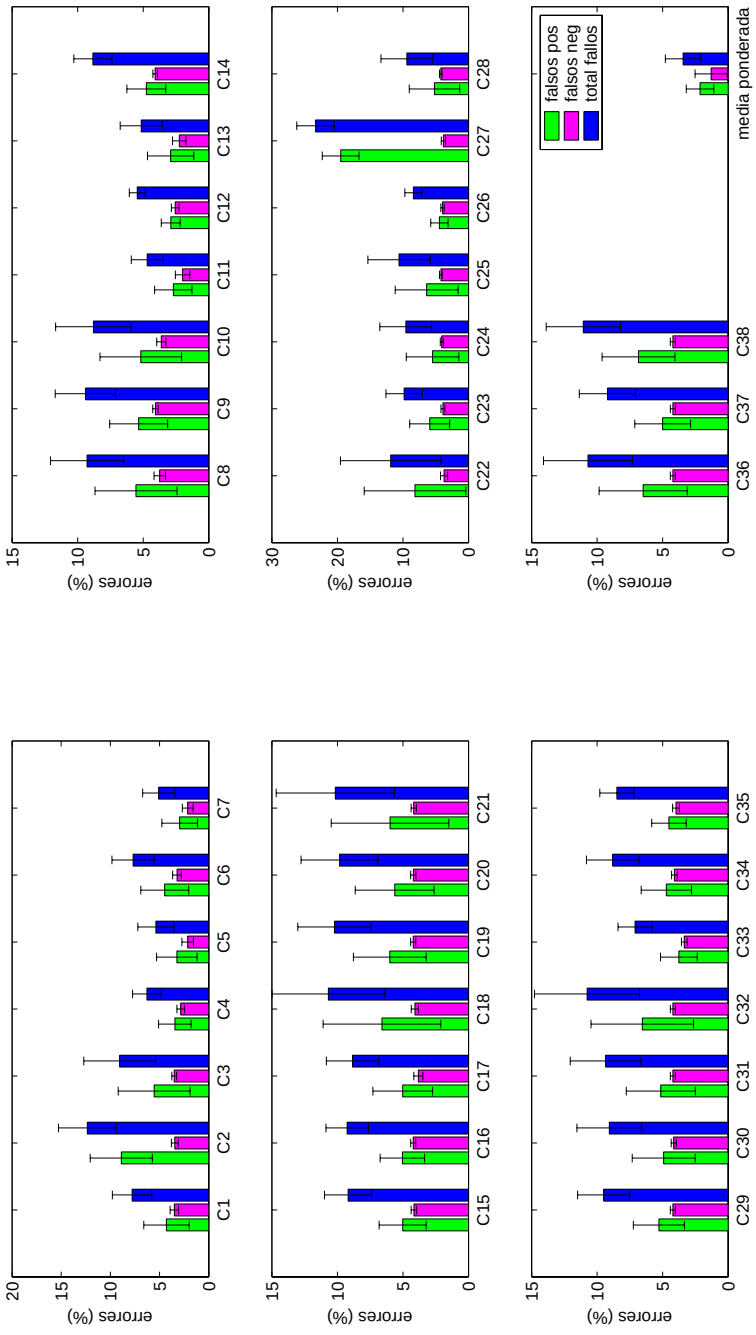


Figura 7.25: Resultados obtenidos para el tercer objeto.

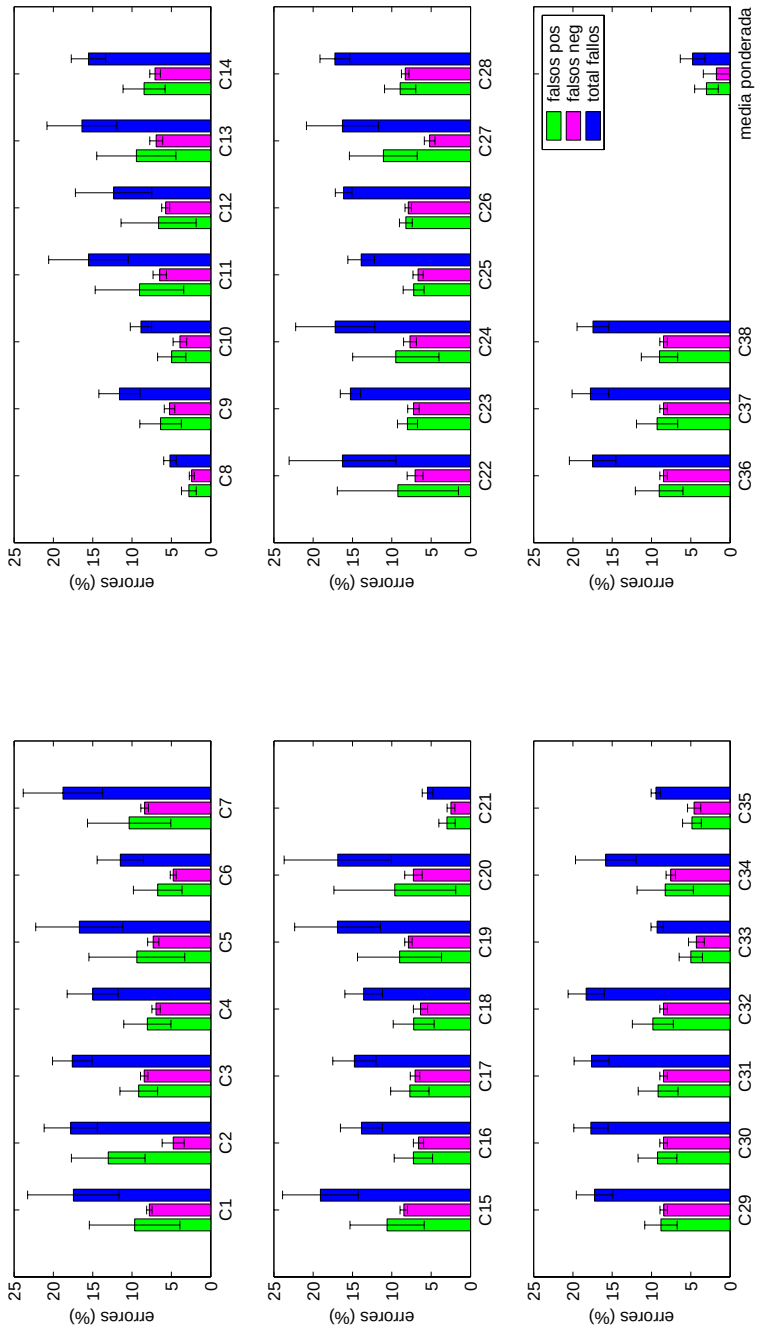


Figura 7.26: Resultados obtenidos para el cuarto objeto.

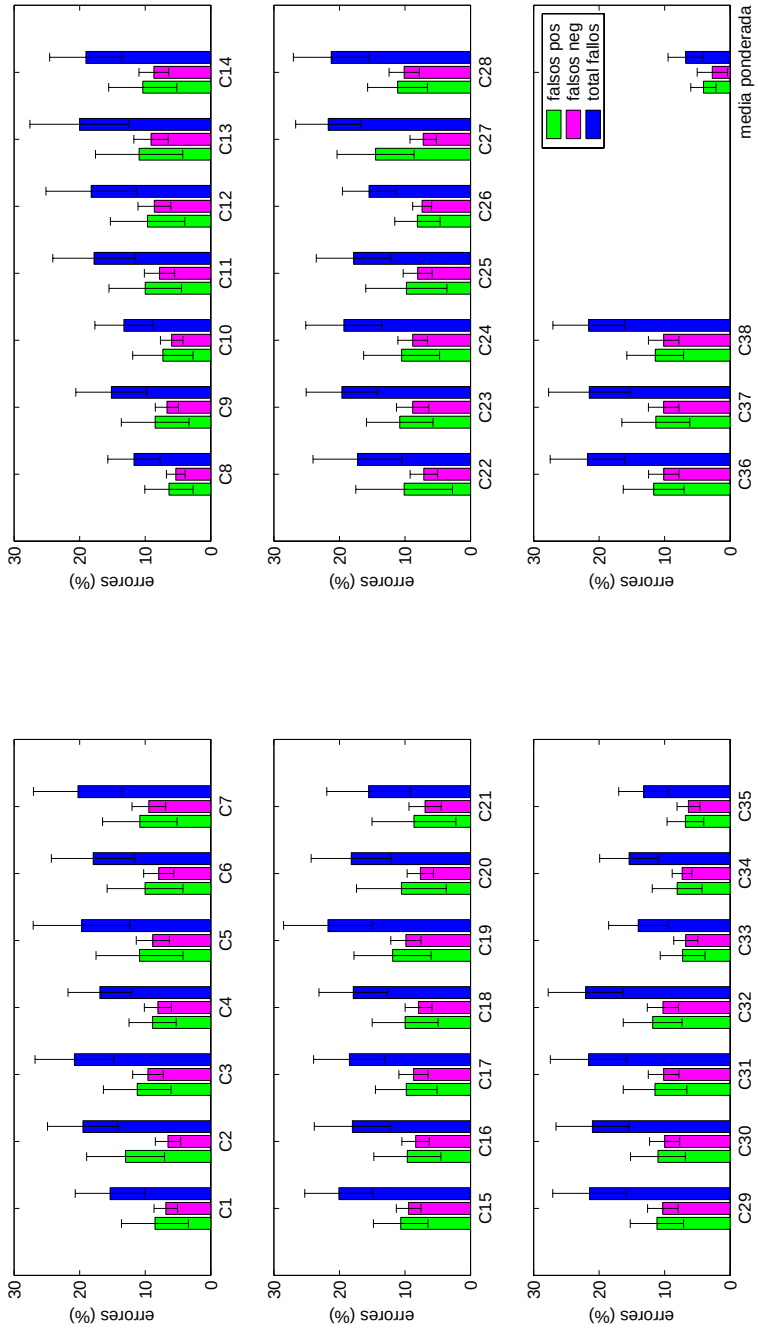


Figura 7.27: Resultados promediados para los cuatro objetos utilizados. Ninguno de los canales es capaz de ofrecer resultados óptimos sobre los cuatro objetos simultáneamente.

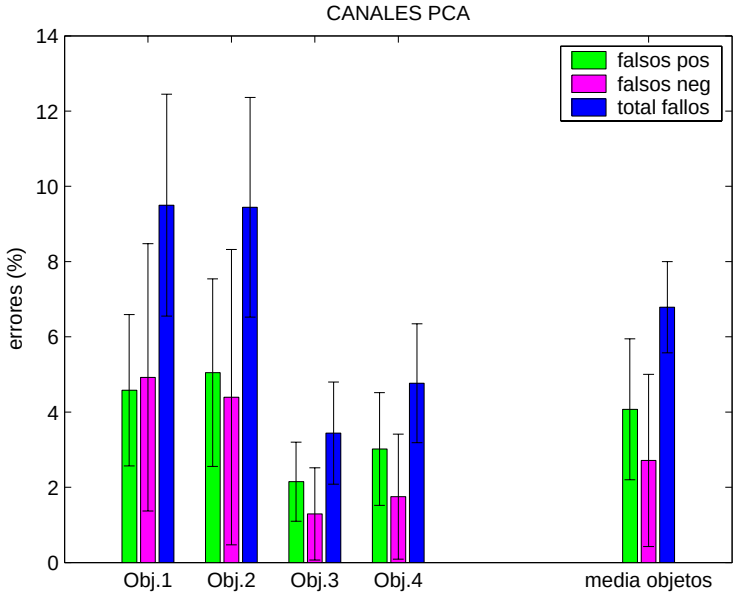


Figura 7.28: Resumen de los resultados obtenidos con la media ponderada de los mapas de probabilidad en la etapa multicanal PCA.

7.4. Capacidad de detección del canal LoG_{ON-OFF} y de la combinación de los procedimientos multicanal y multiescala

Sobre los mismos ejemplos utilizados en el apartado anterior, se ha realizado una comprobación de la capacidad de detección del canal LoG_{ON-OFF} junto con NMFSC, repitiendo exactamente los mismos experimentos de validación cruzada.

Para estas pruebas, se han utilizado escalas en un rango lo suficientemente amplio como para cubrir las distancias existentes entre los objetos y la cámara en cada una de las escenas y se han promediado los mapas de probabilidad, sin ningún tipo de ponderación. El nivel de sparseness (S_W) se ha fijado en 0.7.

Los resultados obtenidos se muestran en la figura 7.29. Puede apreciarse cómo el canal ON-OFF ofrece resultados en las tasas de error ligeramente superiores, en general, a las ofrecidas por la media ponderada de los canales comprimidos mediante PCA; pero normalmente inferiores a las ofrecidas por los canales PCA por separado.

Como última comprobación del funcionamiento del sistema, se muestran los resultados obtenidos al combinar los dos procedimientos de detección, multicanal y mutiescala, mediante la combinación de las zonas umbralizadas obtenidas por ambos procedimientos. Estos resultados se muestran en la figura 7.30, donde puede observarse cómo la combinación de los dos procedimientos consigue reducir el número de píxeles mal clasificados. Lógicamente, dado que la combinación de las zonas umbralizadas se realiza mediante una intersección, el resultado es un menor número de falsos positivos, a costa de incrementar ligeramente los falsos negativos.

En resumen, queda patente el buen funcionamiento del sistema, con unos resultados finales por debajo del $6\,\%$ de píxeles clasificados incorrectamente para el conjunto de escenas estudiadas.

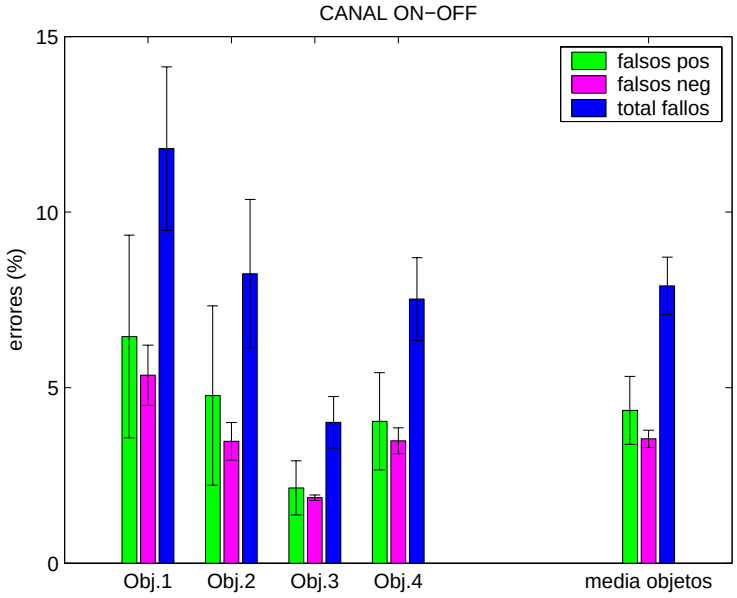


Figura 7.29: Resultados obtenidos con el canal LoG_{ON-OFF} utilizado en la etapa multiescala NMFSC.

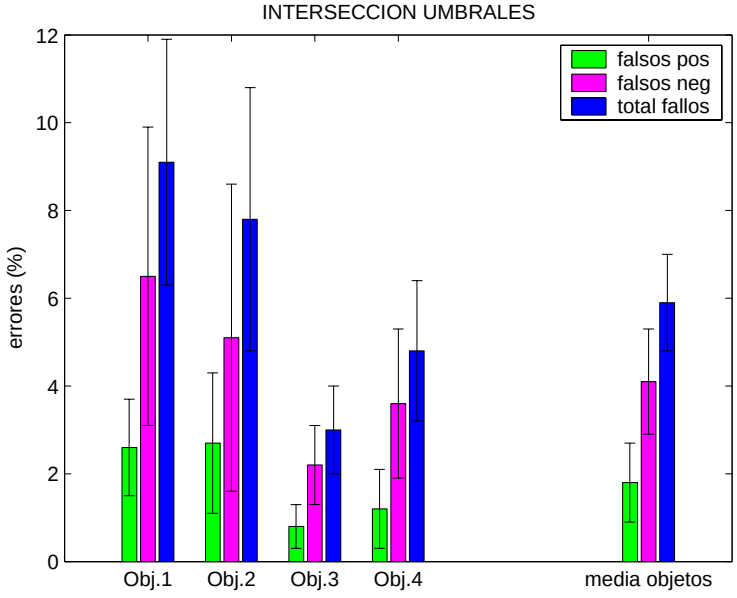


Figura 7.30: Resultados finales tras combinar las etapas multicanal y multiescala.

Capítulo 8

Conclusiones

En el capítulo introductorio se presentaron los proyectos de investigación que constituyen el marco de la presente tesis y los principales objetivos que se perseguían con la misma. En este capítulo de conclusiones se analiza hasta qué punto se han alcanzado los objetivos propuestos, las aportaciones realizadas, su aplicabilidad práctica y los trabajos futuros que han surgido como consecuencia de esta tesis .

8.1. Objetivos alcanzados

Como se ha mostrado a lo largo de este documento, se ha estudiado, analítica y experimentalmente, el comportamiento de algunas de las técnicas de extracción de características lineales más relevantes, dentro del ámbito del reconocimiento de objetos visual basado en la apariencia. La aparente controversia entre los métodos PCA/ICA se ha solucionado, indicando los contextos en los que ICA puede aportar ciertas ventajas frente al método clásico de PCA. Además se han analizado el uso de otros dos métodos (RP y NMFSC) que aportan ventajas al utilizarlos en ciertas aplicaciones (el primer método, disminuye drásticamente el coste computacional implicado en el entrenamiento del sistema y el segundo, permite la obtención de características locales de los objetos a partir de imagenes globales de los mismos).

En base a los resultados obtenidos con el estudio de dichas técnicas se ha propuesto un sistema de detección de objetos basado en la apariencia, que cumple una serie de requisitos o cualidades que le validan como una prometedora metodología para la detección de objetos:

- *Capacidad de trabajar con objetos de forma libre*, debido al uso de características multicanales locales obtenidas a partir de su apariencia.
- *Aplicabilidad en entornos no estructurados*, puesto que los modelos del objeto que se emplean presentan invariancia ante cambios de iluminación y pose del objeto ante la cámara.

- *Posibilidad de trabajar a múltiples escalas*, esto es, la detección del objeto puede realizarse aún cuando el objeto no está siempre a una distancia de la cámara conocida y establecida.

De acuerdo con la metodología expuesta a lo largo del capítulo 6 y los resultados experimentales presentados en el capítulo 7, el sistema de detección propuesto dispone de las cualidades anteriores. Por lo tanto, la metodología de detección de objetos propuesta es válida, y es posible detectar o reconocer un cierto objeto de interés en una escena compleja.

8.2. Aportaciones de la tesis

En la presente tesis se han realizado diferentes aportaciones dentro del campo del reconocimiento visual de objetos. En conjunto, se ha propuesto ***un método de detección de objetos basado en apariencia multicanal y multiescala***, capaz de operar entornos no estructurados, es decir, sin ningún tipo de control en la geometría o apariencia del objeto, la posición del mismo respecto de la cámara, el fondo de la escena, la iluminación del entorno y la presencia de oclusiones o agrupaciones con otros objetos. El desarrollo de este sistema ha requerido analizar en profundidad las arquitecturas de reconocimiento basado en apariencia y las técnicas de extracción de características actuales. El estudio de este campo ha dado lugar a diversas aportaciones puntuales, que a continuación se enumeran:

- Una de las principales aportaciones de esta tesis es la realización de un ***estudio analítico y experimental de las similitudes y diferencias de los métodos de extracción de características más utilizados en la actualidad: PCA e ICA***. Dicho estudio permite aclarar la controversia que existía en el empleo de dichas técnicas, pues como se ha expuesto a lo largo de la tesis, la literatura actual es confusa ante los resultados que proporciona cada uno de los métodos.

  Si no se realiza un proceso de selección de características, ICA y la operación de *whitening* (PCA + escalado) funcionan de manera totalmente equivalente con cualquier tipo de datos si el clasificador utilizado es invariante rotacional. Sin embargo, si se realiza una normalización en el cálculo de las componentes independientes, o bien se realiza una selección de características previa al clasificador, o el clasificador utilizado no es invariante frente a rotaciones, el funcionamiento, y por tanto, las tasas de éxito en el reconocimiento de ambos métodos pueden ser significativamente diferentes.

  Esta aportación no es sólo aplicable al ámbito del reconocimiento visual de objetos basado en la apariencia, sino que es extensible a cualquier sistema de reconocimiento de patrones que haga uso de alguna de estas dos técnicas para la compresión de la información de los patrones de las clases.

- Otra de las aportaciones de la presente tesis es la ***propuesta de aplicabilidad del método RP como método extractor de características para reconocimiento de objetos***. Aunque sus tasas de éxito en el reconocimiento son inferiores a las de otros métodos, presenta otras ventajas como la simplicidad de su generación *off-line* independiente de la naturaleza de los datos. Lo que le hace especialmente útil con bases de datos de objetos a reconocer no estables o variable en el tiempo, esto es, aparecen o desaparecen nuevos elementos durante el funcionamiento del sistema de reconocimiento.

- En este tesis ***se propone una nueva arquitectura basada en la arquitectura objeto original de Murase [118] y que se ha denominado subespacio objeto modificado***. La nueva propuesta permite no sólo detectar orientaciones sino también identificar objetos. De esta manera, los resultados experimentales con PCA realizados con dos librerías de objetos de forma libre (COIL-100 y UMH-15) y que se presentaron en el capítulo 4, muestran que la utilización de estos subespacios objeto modificados resulta beneficiosa para aumentar la capacidad de reconocimiento de los objetos con más variabilidad en sus vistas, en comparación con la utilización de la arquitectura clásica (subespacio universal).

- Otra aportación de la tesis es el ***estudio experimental realizado con un método muy reciente de extracción de características, NMFSC, bajo la arquitectura de reconocimiento propuesta***, siendo su principal ventaja la capacidad para producir representaciones locales de los objetos, lo que permite mejorar el reconocimiento de los objetos en entornos no estructurados, es decir, en presencia de una gran cantidad de objetos, oclusiones o iluminación variable. En los resultados obtenidos en la comparativa realizada entre PCA y NMFSC, se muestra claramente la superioridad de la arquitectura del subespacio objeto modificado junto con el preprocesado de datos propuesto (***el canal*** LoG_{ON-OFF}).

8.3. Aplicabilidad práctica

La metodología de detección de objetos propuesta es completamente novedosa y por tanto su desarrollo se limita al realizado en la presente tesis; esto hace que su aplicabilidad práctica potencial sea muy grande, pero que por el momento sean necesarios desarrollos adicionales para permitir su implantación en sistemas reales de todo tipo.

No obstante, incluso en su estado actual, el sistema es capaz de funcionar correctamente para cualquier objeto con el entrenamiento adecuado, esto es, la generación de los modelos de apariencia PCA y NMFSC; el ajuste de los pesos de los canales y los umbrales de detección proporcionan porcentajes de error en la clasificación de los píxeles de la escena (píxeles clasificados incorrectamente) inferiores al $5\,\%$.

Desde un punto de vista general, las posibles aplicaciones de los desarrollos de la tesis se centran en sistemas robóticos autónomos. Los robots necesitan una cierta percepción del entorno para funcionar con autonomía, y uno de los requerimientos es la capacidad para detectar y reconocer objetos en fondos no estructurados.

El sistema desarrollado en la tesis es particularmente útil en tales escenarios, dado que es capaz de reconocer objetos con independencia del ángulo con el que se capturan las imágenes o de la distancia entre la cámara y el objeto. Podría aplicarse tanto a robots móviles como a brazos robot. En el primero de los casos, puede suponer una ayuda importante para la navegación autónoma; mientras que en el segundo de los casos podría aplicarse, por ejemplo, en procesos de agarres de objetos, ayudando a detectar el objeto de interés y su pose. Para esta última aplicación se requiere conocer con cierta precisión el contorno de los objetos a agarrar; dado que el sistema actualmente sólo ofrece un representación burda de este contorno, se propone utilizar modelos deformables (*snakes*, [25] [89]) para ajustar los contornos a los bordes existentes en la imagen. En el apartado de trabajos futuros se muestran algunos resultados preliminares ya obtenidos.

Un aspecto fundamental que condiciona la aplicabilidad práctica de cualquier sistema de visión artificial es el tiempo de cómputo. El sistema de detección de objetos propuesto, gracias a independendizar el procesamiento multicanal del procesamiento multiescala, consigue mantener un coste computacional reducido aún cuando se utilizan precisiones elevadas en la descomposición de la imagen en ventanas (paso pequeño o, lo que es lo mismo, alto nivel de solapamiento entre ventanas).

No existe ningún procedimiento iterativo, y las operaciones matemáticas a realizar para detectar un objeto son simples multiplicaciones de matrices (proyecciones sobre subespacios generados durante el entrenamiento) y cálculos de distancias (en concreto, se utiliza la distancia euclídea, pero podría utilizarse otra distancia de menor coste computacional como la norma L1)

En un escenario estandar podrían utilizarse ventanas de tamaño 60×60 y un paso de 15 píxeles sobre una escena de tamaño 480×640. Supuestos los 38 canales distintos (ver la tabla A.6), el coste de la detección multicanal en un ordenador estandar (Pentium IV con un 1G de RAM) se sitúa alrededor de $1{,}2$ s. El coste de la detección multiescala, donde sólo hay un canal, se sitúa sobre los 80 ms para la escena de tamaño 480×640 con factor de escalado unidad. Para escalados de la imagen inferiores a la unidad, el tiempo de cómputo disminuye (40 ms para un factor $0{,}5$), mientras que para escalados superiores a la unidad el coste computacional se incrementa (300 ms para un factor de escalado igual a 2). En total, para este caso concreto (PCA con 38 canales y NMFSC con 3 escalados) el coste no supera los 2 segundos.

Si bien estos costes pueden aumentar debido al uso de un mayor número de canales,

la reducción del paso de búsqueda, la disminuión del tamaño de la ventana, el incremento del tamaño de la imagen de la escena o el uso de un mayor número de escalas de la etapa de detección NMFSC, es necesario destacar la facilidad con la que el procedimiento puede ser repartido entre varios procesadores. La facilidad de paralelización hace que el sistema sea aplicable en entornos reales, sin introducir retardos indeseados.

8.4. Trabajos futuros

El sistema de detección visual de objetos desarrollado en la presente tesis deja abierto el camino para multitud de trabajos futuros. Entre ellos cabe destacar:

- *Experimentos de selección de características con ICA.* En el capítulo 5 de la tesis se demuestra teóricamente y experimentalmente la equivalencia entre ICA y PCA como métodos de compresión de la información (o métodos de extracción de características) cuando se cumplen dos condiciones fundamentales: utilización de todas las componentes y aplicación de clasificadores invariantes ante rotaciones. En el caso de realizar selección de características, los resultados obtenidos por PCA e ICA pueden ser diferentes. Se plantea como trabajo futuro el estudio tanto experimental como teórico de posibles procedimientos de selección de características con ICA (con PCA la selección más idónea suele ser la elección de componentes de máxima a mínima varianza, salvo algunas excepciones en el caso de aplicaciones de reconocimiento de objetos donde las clases son muy cercanas, como en el reconocimiento de caras, donde se suelen descartar las 3 o 4 primeras componentes y mantener el resto ordenadas de máxima a mínima varianza). Actualmente no existen criterios que permitan determinar cuáles de las componentes independientes consiguen una mejor separación entre clases; el objetivo es, por una parte, buscar criterios plausibles; y por otra, determinar si la selección de características sobre ICA permite mejorar los resultados obtenidos al realizar selección de características sobre PCA.

- *Implementación en paralelo.* De acuerdo con lo comentado anteriormente, el sistema de detección de objetos propuesto permite paralelizar múltiples procesamientos, de forma que el tiempo de cómputo se puede reducir apreciablemente. Como trabajo futuro se considera la implementación del detector en un sistema multiprocesador o bien en un entorno de red. Caben dos estrategias de paralelización: o bien procesar cada canal o grupo de canales y cada escala o grupo de escalas en procesadores diferentes para posteriormente promediar los resultados obtenidos; o bien procesar cada ventana o grupo de ventanas en equipos diferentes, para luego combinar los resultados correspondientes a cada pixel. El resultado sería un sistema de detección que permitiría compaginar un alto nivel de precisión (paso pequeño entre ventanas y muchas escalas) con una velocidad de procesado muy elevada (funcionamiento en tiempo real).

- *Aplicación a problemas de categorización.* Otro posible trabajo futuro sobre los desarrollos de la tesis es la aplicación a problemas de categorización o catalogación, en los cuales el objetivo no es detectar un cierto objeto específico sino detectar cualquier objeto perteneciente a una cierta clase. En principio, el método presentado en la tesis sería aplicable directamente a este tipo de problemas, y sólo sería necesario modificar el tipo de imágenes de entrenamiento suministradas (en lugar de múltiples vistas de un cierto objeto, múltiples vistas de varios objetos pertenecientes a una clase determinada). La principal diferencia con el problema de detección es la variación en la influencia o peso de los canales.

- *Extracción del contorno de los objetos detectados.* En función de la aplicación sobre la que se desea utilizar el detector de objetos desarrollado, puede ser necesario determinar el contorno de los objetos detectados. Por ejemplo, para una aplicación de agarre de objetos con la mencionada en el apartado anterior, no basta con determinar la zona donde se encuentra el objeto en la escena, sino que se necesita conocer el contorno exterior del mismo. En este sentido, se ha iniciado una línea de trabajo basada en la utilización de modelos deformables o *snakes* [25] [89]. Tales modelos se inicializan con la forma de la región devuelta por el detector de objetos (unión de píxeles cuyo valor de probabilidad supera un determinado umbral) y que se deforma de acuerdo con una función potencial que tiende a aproximar el contorno a los bordes presentes en la imagen (extraídos con un detector de bordes estandar como por ejemplo el detector de Canny).

Apéndice A

Notación

Las siguientes tablas indican los nombres de variables más relevantes y los convenios de notación utilizados en cada unos de los capítulos de la presente tesis.

$I(x, y)$, $I(f, c)$	Una imagen 2D
f	Número de filas de píxeles en la imagen
c	Número de columnas de píxeles en la imagen
$M = f \times c$	Dimensión de la imagen
$\mathbf{x}$	Vector relativo a la imagen $I(x, y)$ utilizando el mapeado de la ecuación 2.2
$\widetilde{\mathbf{x}}$	Vector relativo a una nueva imagen
N	Número de ejemplos o imágenes
K	Número de ejemplos o imágenes prototipo de cada clase (u objeto)

Tabla A.1: Notación utilizada en el capítulo 2.

$\mathbf{x}$	Representación vectorial del patrón o variable aleatoria (relativo a $I(x,y)$)
M	Número de componentes del vector $\mathbf{x}$
N	Número de ejemplos de patrones
$\mathbf{X}$	Espacio de patrones o matriz conjunto de todos los N patrones, $\mathbf{X} \in \Re^{M \times N}$
P	Dimensión reducida del patrón $\mathbf{x}$, $P \leq M$
$W()$	Función general de transformación de extracción de características
$\mathbf{W}^T$	Matriz de transformación de extracción de características lineal
J	Número de clases distintas
ω_j	Clase
ω_0	Clase rechazo
Ω	Conjunto de las clases
$\mathbf{u}$	Dirección de $\Re^M$, en PCA esta dirección es la de máxima varianza
p_j	Proyección del patrón sobre una dirección $\mathbf{u}$
$\mathbf{p}$	Proyección del patrón sobre un subespacio $\mathbf{U}$
$\mathbf{P}$	Matriz de proyecciones, cada columna es la proyección de un patrón
$\mathbf{C_X}$	Matriz de covarianzas de $\mathbf{X}$
λ	Valor propio
$\mathbf{U}$	Matriz que contiene en sus columnas a los vectores propios de $\mathbf{X}$
$\mathbf{\Lambda}$	Matriz que contiene en la diagonal a los valores propios de $\mathbf{X}$
$\mathbf{y}$	Patrón reconstruido a partir de $\mathbf{p}$ y $\mathbf{U}$
$\mathbf{S_W}$	Matriz de covarianza interna media *(within-class scatter matrix)*
$\mathbf{S_b}$	Matriz de covarianza entre las clases *(between-class scatter matrix)*
$\mathbf{J^{PCA}}$	Función a optimizar en PCA
$\mathbf{J^{LDA}}$	Función a optimizar en LDA
$\mathbf{s}$	Vector fuente o componentes independiente (IC)
$\mathbf{A}$	Matriz mezcla, $\mathbf{A} \in \Re^{M \times N}$
$\mathbf{n}$	Vector representativo del ruido aditivo
$\mathbf{B}$	Matriz ortogonal que maximiza la no-gaussianidad
$G()$	Función de estimación del grado de no-gaussianidad
$\mathbf{b}$	Proyección no-gaussiana
κ	Kurtosis
$H()$	Entropía
$N()$	Negentropía
$p()$	Densidad de probabilidad
E	Esperanza
$\mathbf{R}$	Matriz de números aleatorios
$\mathbf{W}$, $\mathbf{H}$	Matrices no negativas resultado de la descomposición NMF
S_w, S_h	Grados de raleza (sparseness) de $\mathbf{W}$ y $\mathbf{H}$

Tabla A.2: Notación utilizada en el capítulo 3.

$\mathbf{x}_j^t$	Vector patrón (imagen) j del objeto t
T	Número de objetos diferentes
G_t	Conjunto de imágenes de entrenamiento del objeto t
G_U	Conjunto de imágenes universal
$\mathbf{P}_t$	Matriz de proyecciones del objeto t
$\mathbf{W}_t^T$	Matriz de transformación del objeto t (subespacio objeto propio)
$\mathbf{T}_{Wt}$	Proyecciones de un conjunto de imágenes en cada uno de los subespacios objeto propios
$\mathbf{W}_{om}^T$	Matriz de transformación del subespacio objeto modificado
$\mathbf{P}_{om}$	Matriz de proyecciones en el subespacio objeto modificado
$O()$	Orden del coste computacional
$L2$	Distancia euclídea o norma L2
MAH	Distancia de Mahalanobis

Tabla A.3: Notación utilizada en el capítulo 4.

$\mathbf{X}$	Espacio de patrones, $\mathbf{X} \in \Re^{M \times N}$
$\mathbf{U}$	Matriz ortonormal, $\mathbf{U} \in \Re^{M \times M}$
$\widetilde{\mathbf{U}}$	Matriz ortonormal reducida, $\widetilde{\mathbf{U}} \in \Re^{K \times M}$
$\widetilde{\mathbf{Y}}$	Espacio de proyecciones reducidas, $\widetilde{\mathbf{Y}} \in \Re^{K \times N}$
$\mathbf{V}$	Matriz ortonormal (transformacion SVD)
$\mathbf{\Lambda}$	Matriz ortonormal (transformacion SVD)
$\widetilde{\mathbf{Z}}$	Matriz de datos blanqueados (transformación de *whitening*)
$\mathbf{S}$	Matriz de las fuentes o componentes independientes
$\mathbf{A}$	Matriz mezcla
$\mathbf{W}$	Matriz de separación (pseudoinversa de $\mathbf{A}$)
$\mathbf{R}$	Matriz de rotación (búsqueda de la no-gaussianidad)
α_t	Índice de separabilidad trivial del objeto t
β_t	Índice de separabilidad medio del objeto t
$\mathbf{X_p}$	Datos positivos de $\mathbf{X}$
$\mathbf{X_n}$	Datos negativos de $\mathbf{X}$
$\mathbf{X}_{ON-OFF}$	Datos en el canal ON-OFF
L	Operador laplaciana
LoG	Operador laplaciana de la gaussiana

Tabla A.4: Notación utilizada en el capítulo 5.

$\mathbf{P}_i^*$	Modelo de apariencia del objeto de interés en el canal i (PCA)
N	Número total de canales del sistema
$\mathbf{v}_k$	Ventana k en la escena
$(\mathbf{p}_k)_i$	Proyección de la ventana centrada k en el canal i
$\mathbf{\Lambda}$	Matriz de valores propios (subespacio PCA)
$\mathbf{U}$	Matriz de vectores propios (subespacio PCA)
$\mathcal{D}_i$	Mapa de distancias en el canal i
$\mathcal{M}_i$	Mapa de probabilidades en el canal i
$\mathcal{M}$	Mapa de probabilidades global
$\mathbf{P}_{NMFSC}^*$	Modelo de apariencia del objeto de interés en el canal ON-OFF (NMFSC)
$I(C,F)$	Información mutua entre el fragmento F y la clase C
H	Entropía
N_v	Número total de ventanas de la escena en la etapa de búsqueda
N_x, N_y	Número de píxeles de la imagen
w_x, w_y	Tamaño de la ventana rectangular en píxeles
p_x, p_y	Paso del barrido en ambos ejes
c_{ij}	Correlación de la escena j en el canal i
K_i	Fiabilidad o peso del canal i
γ	Factor de escalado
$\widetilde{\mathcal{M}}_i$	Mapa de probabilidades transformado en el canal i
ν_i	Umbral del canal i
ν_G	Umbral global

Tabla A.5: Notación utilizada en el capítulo 6.

$C1$	Canal de apariencia H
$C2$	Canal de apariencia S
$C3$	Canal de apariencia V
$C4$	Canal de apariencia I
$C5$	Canal de apariencia r
$C6$	Canal de apariencia g
$C7$	Canal de apariencia b
$C8$	Canal de apariencia l_1
$C9$	Canal de apariencia l_2
$C10$	Canal de apariencia l_3
$C11$	Canal de apariencia c_1
$C12$	Canal de apariencia c_2
$C13$	Canal de apariencia c_3
$C14$	Canal de histogramas RGB
$C15$	Canal de histogramas H
$C16$	Canal de histogramas r
$C17$	Canal de histogramas g
$C18$	Canal de histogramas b
$C19$	Canal de histogramas S
$C20$	Canal de histogramas l_1
$C21$	Canal de histogramas l_2
$C22$	Canal de histogramas l_3
$C23$	Canal de histogramas c_1
$C24$	Canal de histogramas c_2
$C25$	Canal de histogramasc_3
$C26$	Canal de forma H
$C27$	Canal de forma S
$C28$	Canal de forma V
$C29$	Canal de forma I
$C30$	Canal de forma r
$C31$	Canal de forma g
$C32$	Canal de forma b
$C33$	Canal de forma l_1
$C34$	Canal de forma l_2
$C35$	Canal de forma l_3
$C36$	Canal de forma c_1
$C37$	Canal de forma c_2
$C38$	Canal de forma c_3

Tabla A.6: Notación utilizada en el capítulo 7 relativa a los canales utilizados.

Apéndice B

Algoritmo NMFSC

B.1. Algoritmo NMFSC

El algoritmo NMFSC de Hoyer [75] se basa en un algoritmo de gradiente descendente proyectado [79] con restricciones de *sparseness* en el operador de proyección. Este algoritmo avanza hacia la dirección del gradiente descendente, y a continuación realiza la proyección sobre el espacio restringido. La fuerza de este algoritmo radica en el operador de proyección utilizado, el cual fuerza el grado de *sparseness* deseado. Este operador se describe posteriormente. El algoritmo NMFSC se detalla a continuación:

1 Inicializar $\mathbf{W}$ y $\mathbf{H}$ como matrices de números positivos aleatorios

2 Si las restricciones de *sparseness* se aplican sobre $\mathbf{W}$, entonces proyectar cada columna de $\mathbf{W}$ de manera que sus componentes sean no-negativos, su norma L2 no cambie, pero si su norma L1 con el fin que se acerce al grado de *sparseness* deseado.

3 Si las restricciones de *sparseness* se aplican sobre $\mathbf{H}$, entonces proyectar cada fila de $\mathbf{H}$ de manera que sus componentes sean no-negativos, su norma L2 no cambie, pero si su norma L1 con el fin que se acerce al grado de *sparseness* deseado.

4 Iterar

(a) Si las restricciones de *sparseness* se aplican sobre $\mathbf{W}$

i Hacer $\mathbf{W} := \mathbf{W} - \mu_{\mathbf{W}}(\mathbf{WH} - \mathbf{X})\mathbf{H}^T$

ii Proyectar cada columna de $\mathbf{W}$ de manera que sus componentes sean no-negativos, su norma L2 no cambie, pero si su norma L1 con el fin que se acerce al grado de *sparseness* deseado.

si no, realizar la operación $\mathbf{W} := \mathbf{W} \otimes (\mathbf{XH}^T) \oslash (\mathbf{WHH}^T)$

(b) Si las restricciones de *sparseness* se aplican sobre $\mathbf{H}$

i Hacer $\mathbf{H} := \mathbf{H} - \mu_{\mathbf{H}}\mathbf{W}^T(\mathbf{WH} - \mathbf{X})$

ii Proyectar cada fila de $\mathbf{H}$ de manera que sus componentes sean no-negativos, su norma L2 no cambie, pero si su norma L1 con el fin que se acerce al grado de *sparseness* deseado.

si no, realizar la operación $\mathbf{H} := \mathbf{H} \otimes (\mathbf{W}^T\mathbf{X}) \oslash (\mathbf{W}^T\mathbf{W}\mathbf{H})$

En el algoritmo anterior, los operadores $\otimes$ y $\oslash$ denotan la multiplicación y división componente a componente entre dos matrices de igual tamaño. Las constantes $\mu_{\mathbf{W}}$ y $\mu_{\mathbf{H}}$ son parámetros que deben ser ajustados adecuadamente para el correcto funcionamiento del algoritmo. En [75] estos parámetros se ajustan automáticamente en función de los datos de entrada. Los pasos multiplicativos a realizar cuando no se aplican las restricciones están tomados directamente del algortimo original de NMF de Lee y Seung [97].

B.2. Operador de proyección

En muchos de los pasos del algoritmo NMFSC se requiere un operador de proyección que fuerce el grado de *sparseness* deseado mediante el ajuste de las normas L1 y L2 del vector en cuestión. Este operador se define en [75] de la siguiente forma:

definición Dado un vector $\mathbf{x}$, encontrar el vector $\mathbf{s}$ no-negativo más cercano (en el sentido euclídeo) con unas norma L_1 y L_2 dadas.

algoritmo El siguiente algoritmo resuelve el problema anterior:

1 Realizar $s_i := x_i + (L_1 - \sum x_i)/dim(\mathbf{x})$

2 Hacer $Z := \{\}$

3 Iterar

(a) Realizar
$m_i := L_1/(dim(\mathbf{x}) - size(Z))$, si $i \notin Z$
$m_i := 0$, si $i \in Z$

(b) Realizar $\mathbf{s} := \mathbf{m} + \alpha(\mathbf{s} - \mathbf{m})$, donde $\alpha \geq 0$ es seleccionada para que el vector $\mathbf{s}$ resultante satisfaga el valor de L_2 impuesto. Esto supone resolver una ecuación cuadrática.

(c) Si todas las componentes de $\mathbf{s}$ son no-negativas, devolver $\mathbf{s}$ y finalizar.

(d) Hacer $Z := Z \cup \{i; s_i < 0\}$

(e) Hacer $s_i := 0, \forall i \in Z$

(e) Calcular $c := (\sum s_i - L_1)/(dim(\mathbf{x}) - size(Z))$

(f) Hacer $s_i := s_i - c, \forall i \notin Z$

(g) Ir al paso (a)

En `http://www.cs.helsinki.fi/hiit_bru/neuro_stat_nmfsc.html` está disponible el código matlab que implementa el algoritmo NMFSC con el operador de proyección antes descrito.

Apéndice C

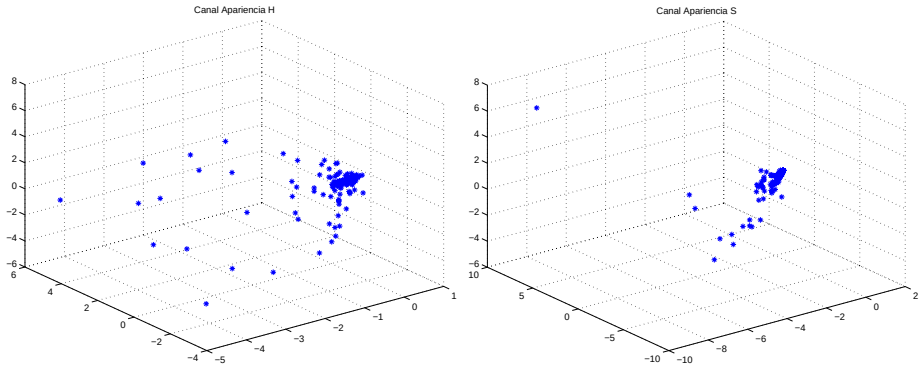
Modelos y distribuciones espaciales del objeto *caja amarilla*

C.1. Modelos de apariencia del objeto *caja amarilla*

En esta sección se muestran los modelos de apariencia del objeto *caja amarilla* en todos los canales de información que utiliza el sistema de detección propuesto. Estos modelos son la representación del objeto utilizada por el sistema de detección para la búsqueda del objeto de interés en una escena compleja.

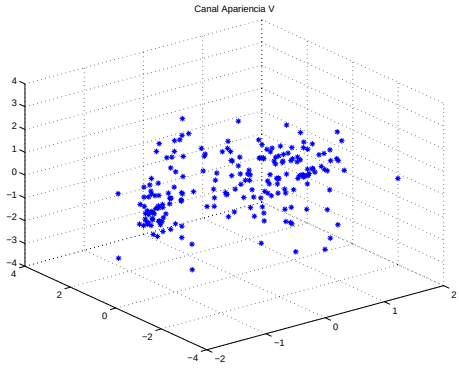
Básicamante, los modelos son equivalentes a los fragmentos (o a las imágenes completas de entrenamiento) pero en un espacio de dimensiones más reducido gracias al uso de alguna técnica de extracción de características.

A continuación, en las siguientes figuras, se muestra a modo de ejemplo los modelos de apariencia de la *caja amarilla* haciendo uso sólo de tres dimensiones.

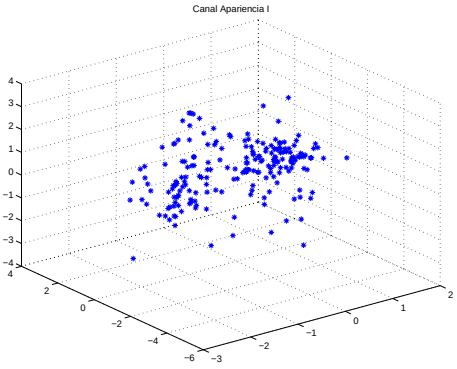


(a) Canal Apariencia H

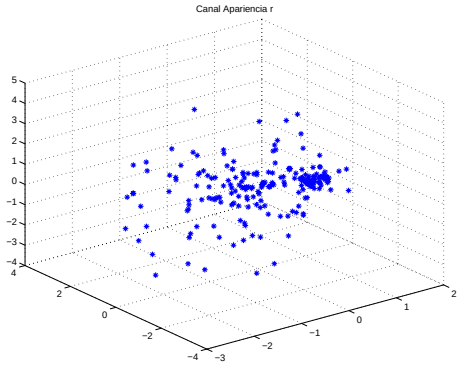
(b) Canal Apariencia S



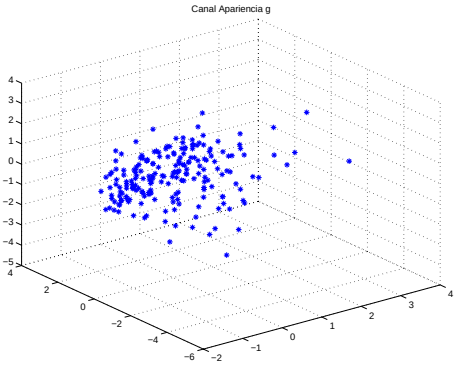
(c) Canal Apariencia V



(d) Canal Apariencia I

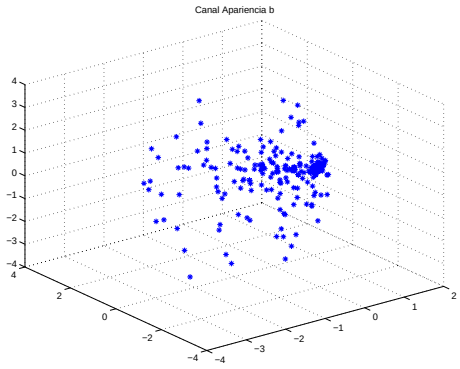


(e) Canal Apariencia r

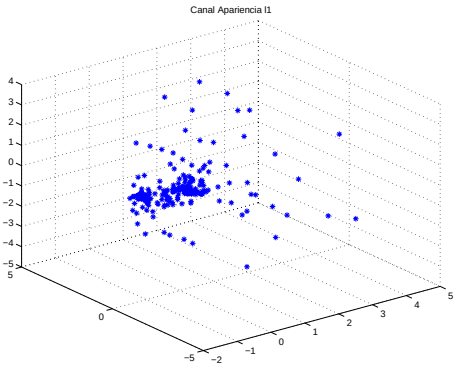


(f) Canal Apariencia g

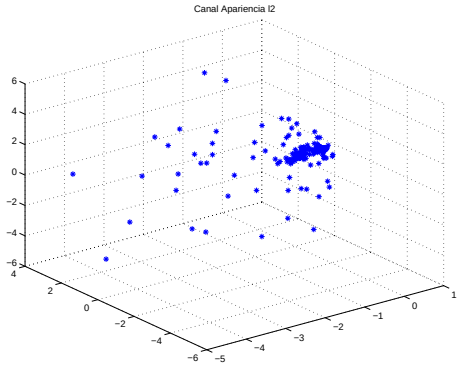
Figura C.1: Modelos del objeto *la caja amarilla* en los canales de Apariencia H, S, V, I, r y g.



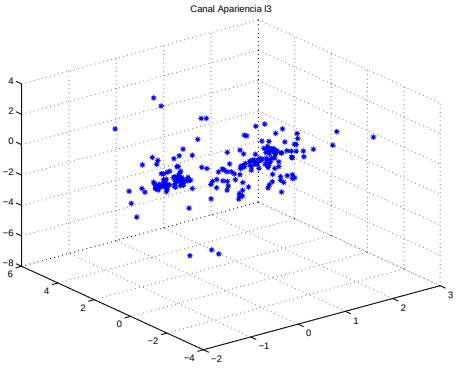
(a) Canal Apariencia b



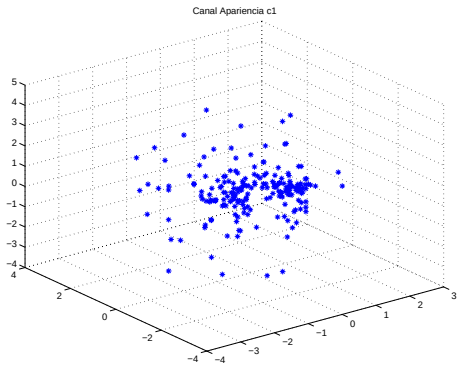
(b) Canal Apariencia l1



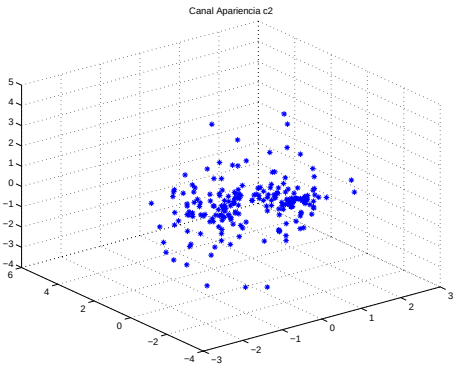
(c) Canal Apariencia l2



(d) Canal Apariencia l3

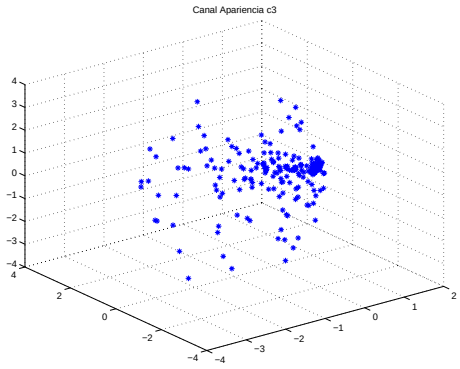


(e) Canal Apariencia c1

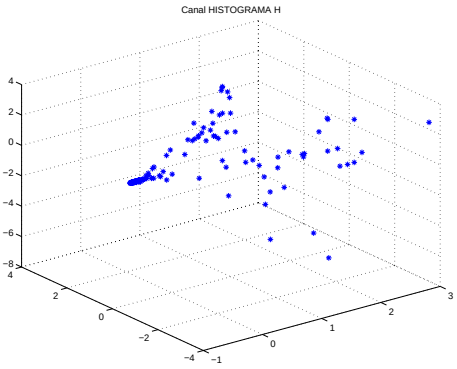


(f) Canal Apariencia c2

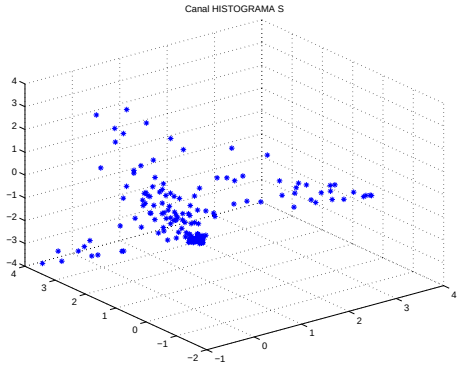
Figura C.2: Modelos del objeto *la caja amarilla* en los canales de Apariencia b, l1, l2, l3, c1 y c2.



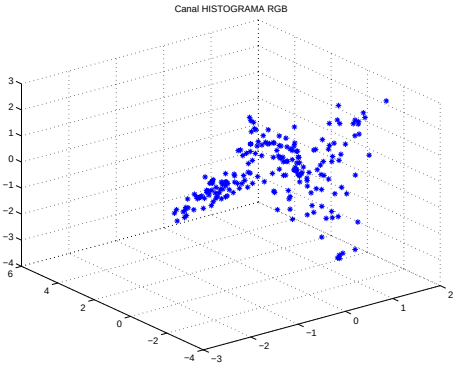
(a) Canal Apariencia c3



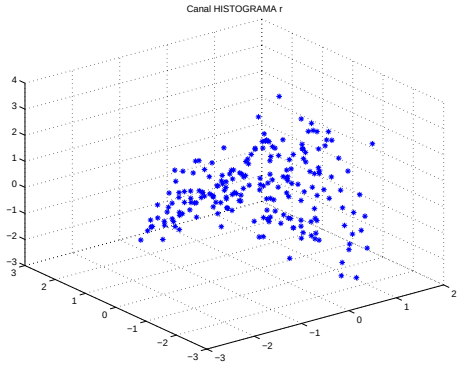
(b) Canal Histograma H



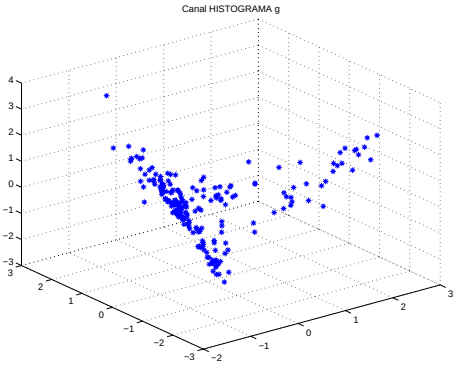
(c) Canal Histograma S



(d) Canal Histograma RGB

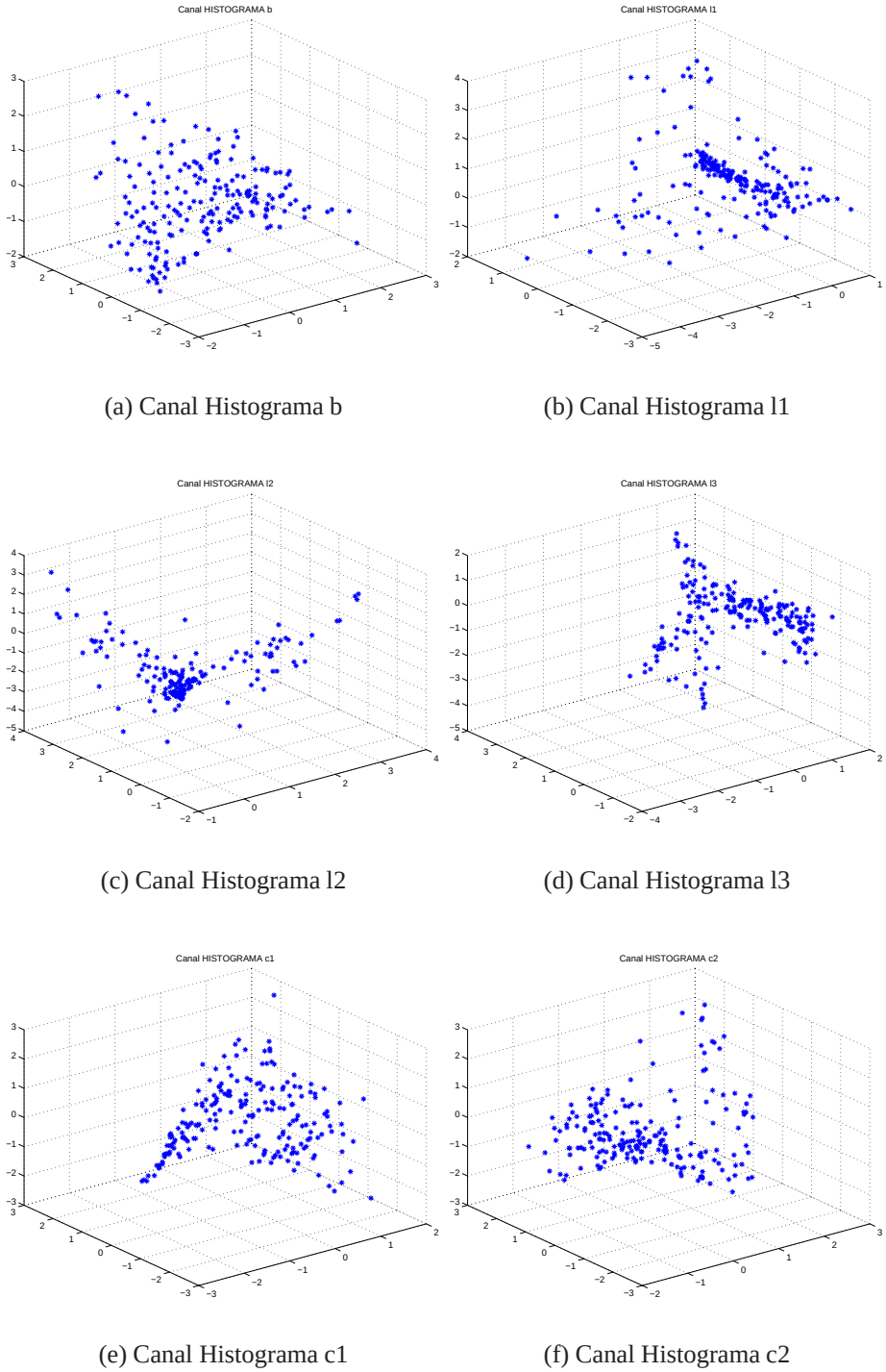


(e) Canal Histograma r



(f) Canal Histograma g

Figura C.3: Modelos del objeto *la caja amarilla* en el canal de Apariencia c3 y en los canales de Histograma H, S, RGB, r y g.



(a) Canal Histograma b

(b) Canal Histograma l1

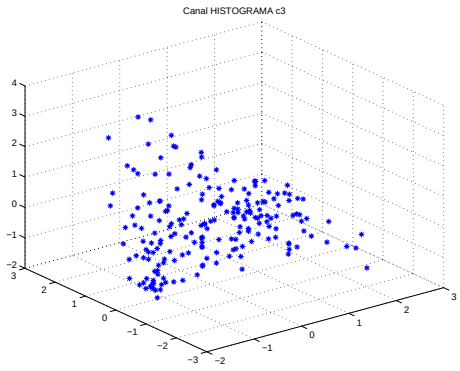
(c) Canal Histograma l2

(d) Canal Histograma l3

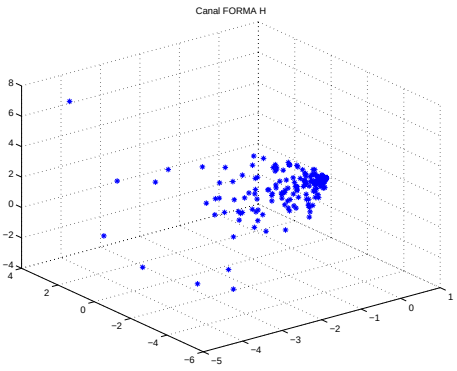
(e) Canal Histograma c1

(f) Canal Histograma c2

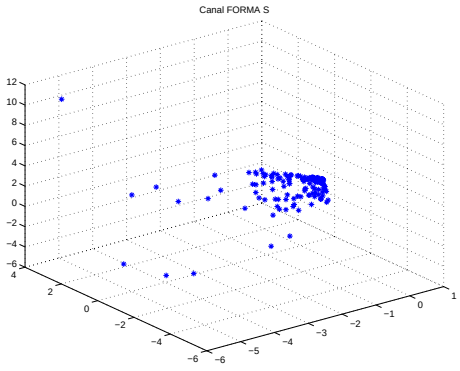
Figura C.4: Modelos del objeto *la caja amarilla* en los canales de Histograma b, l1, l2, l3, c1 y c2.



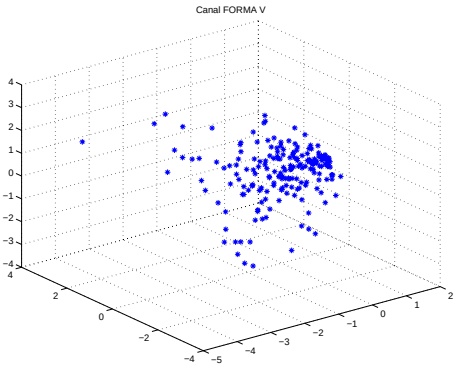
(a) Canal Histograma c3



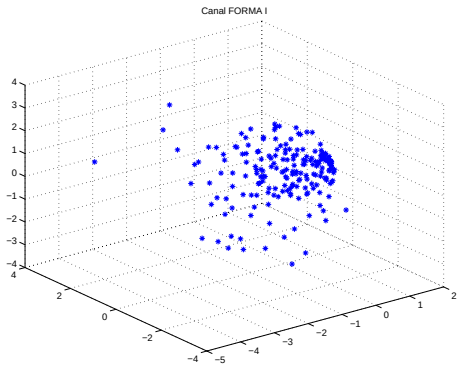
(b) Canal Forma H



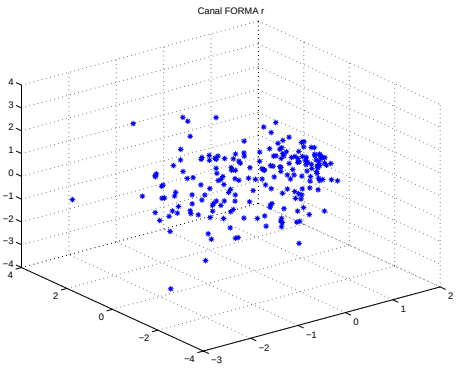
(c) Canal Forma S



(d) Canal Forma V

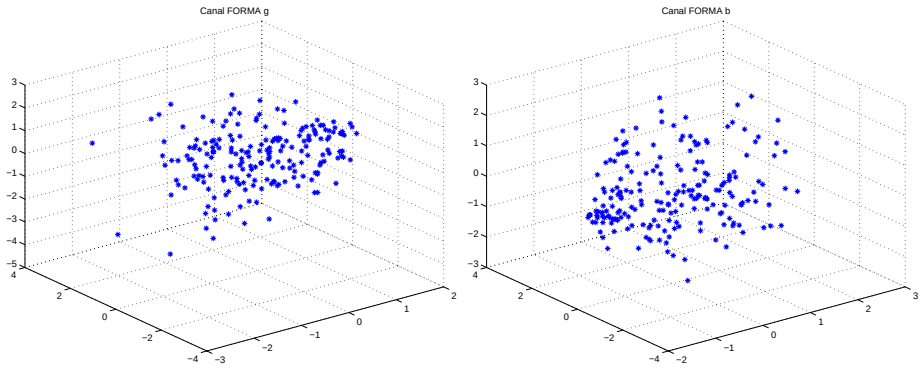


(e) Canal Forma I



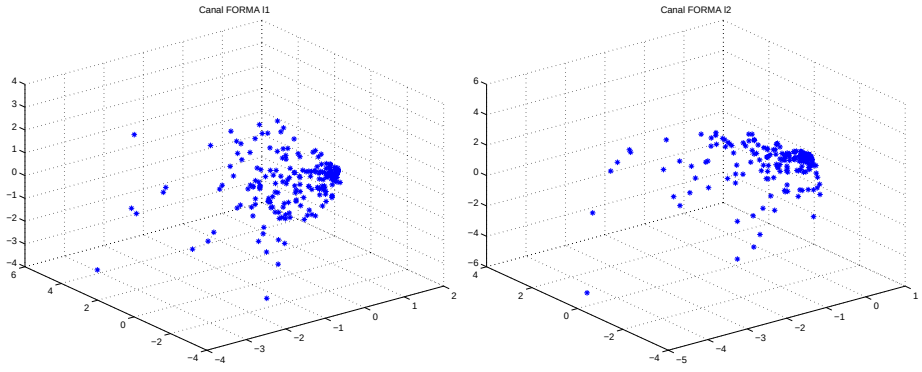
(f) Canal Forma r

Figura C.5: Modelos del objeto *la caja amarilla* en el canal de Histograma c3 y en los canales de Forma H, S, V, I y r.



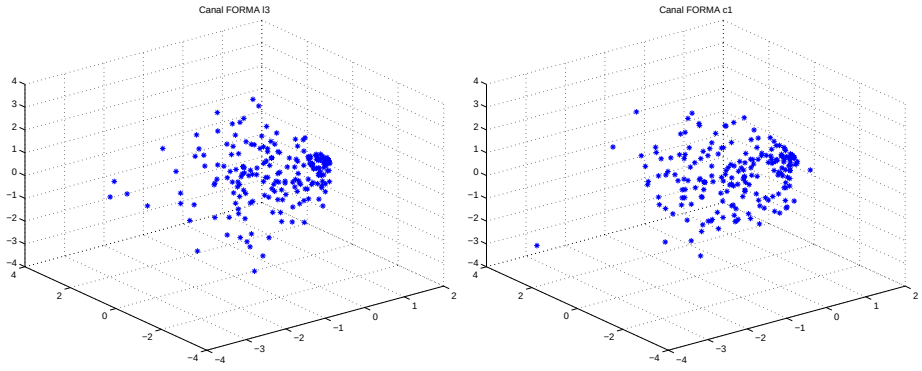
(a) Canal Forma g

(b) Canal Forma b



(c) Canal Forma l1

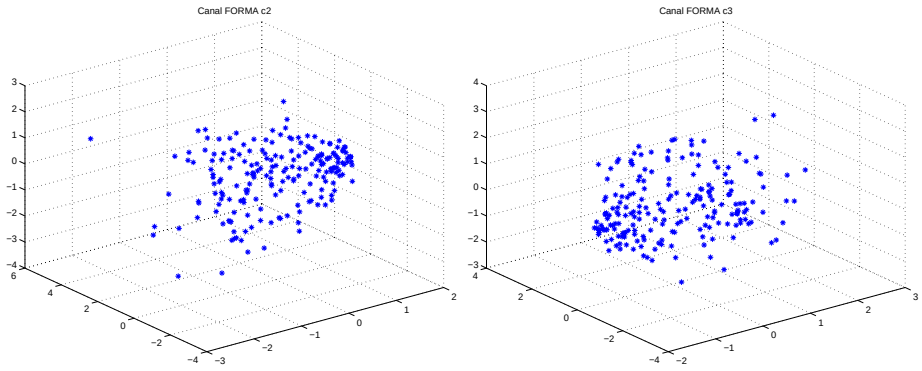
(d) Canal Forma l2



(e) Canal Forma l3

(f) Canal Forma c1

Figura C.6: Modelos del objeto *la caja amarilla* en los canales de Forma g, b, l1, l2, l3 y c1.



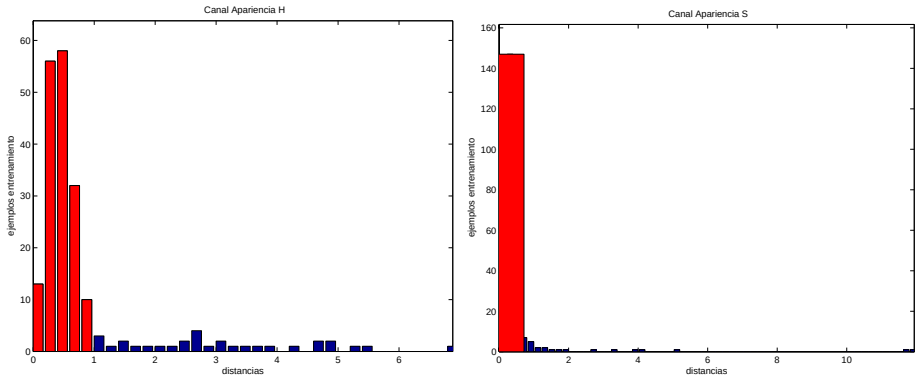
(a) Canal Forma c2 (b) Canal Forma c3

Figura C.7: Modelos del objeto *la caja amarilla* en los canales de Forma c2 y c3.

C.2. Distribuciones espaciales de los modelos de apariencia del objeto *caja amarilla*

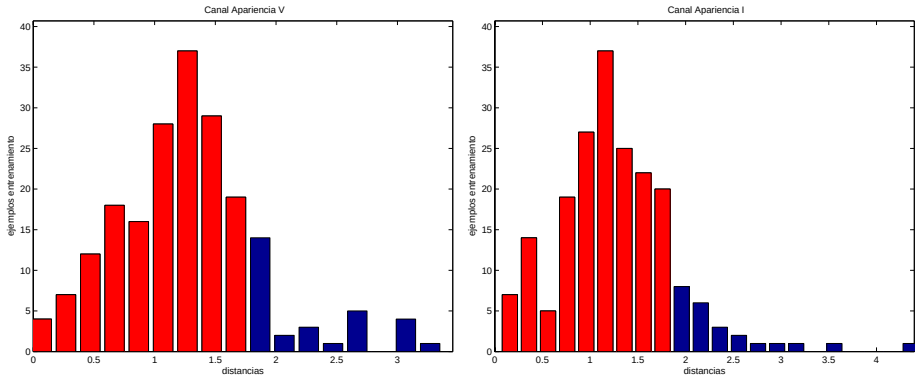
La distribución espacial del modelo de apariencia en un cierto canal puede utilizarse para determinar de modo automático un umbral de reconocimiento tal como se ha visto en el capítulo 6.

A continuación, en las siguientes figuras, se muestra a modo de ejemplo las distribuciones espaciales correspondientes a los modelos de apariencia de la *caja amarilla* mostrados en las figuras anteriores.



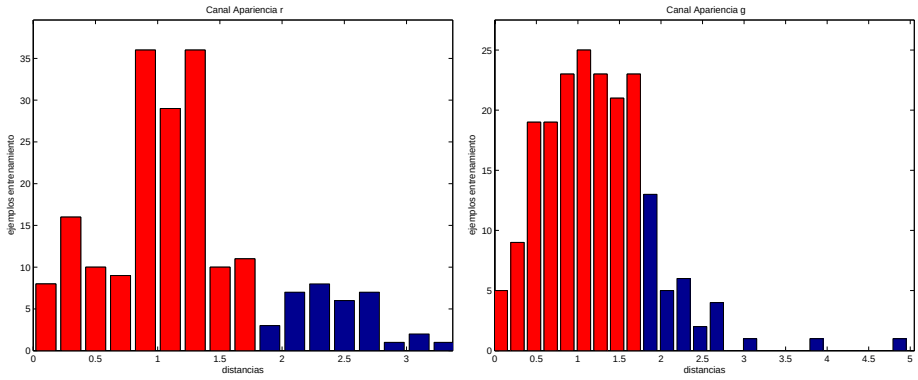
(a) Canal Apariencia H

(b) Canal Apariencia S



(c) Canal Apariencia V

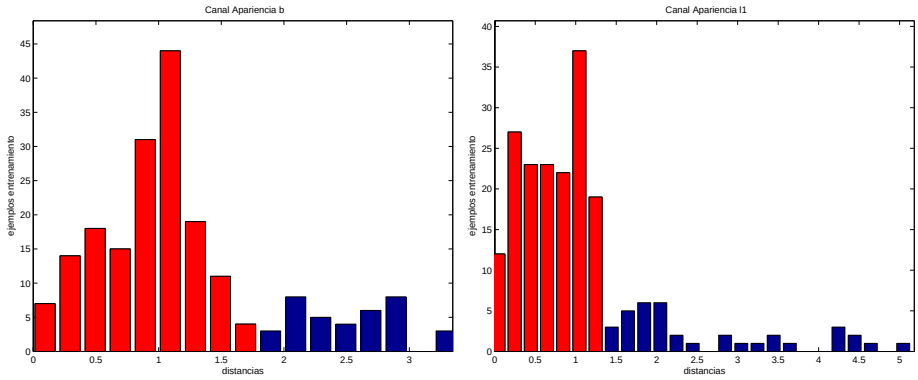
(d) Canal Apariencia I



(e) Canal Apariencia r

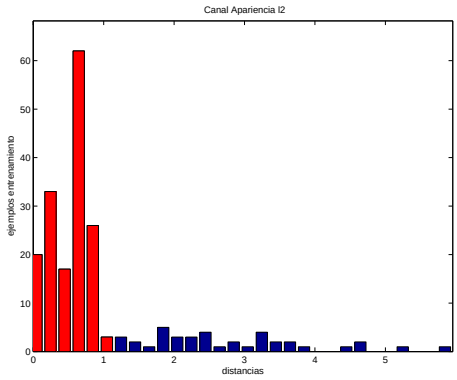
(f) Canal Apariencia g

Figura C.8: Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Apariencia H, S, V, I, r y g..

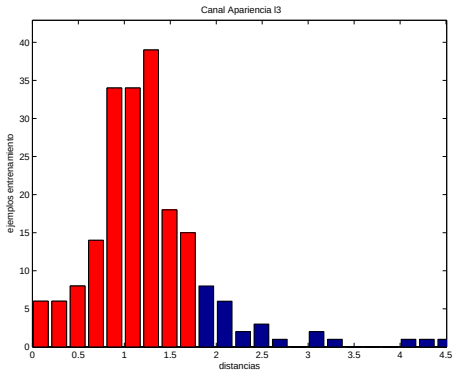


(a) Canal Apariencia b

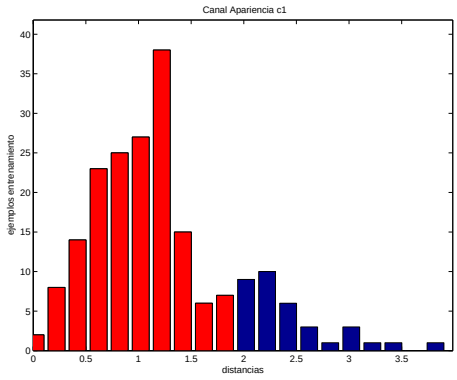
(b) Canal Apariencia l1



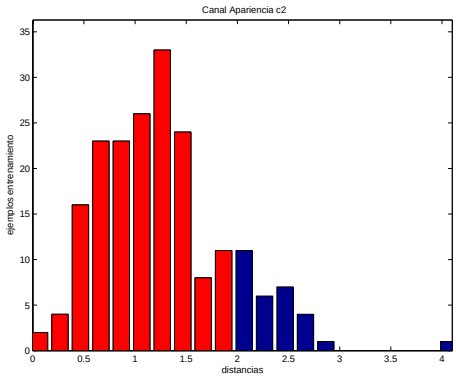
(c) Canal Apariencia l2



(d) Canal Apariencia l3

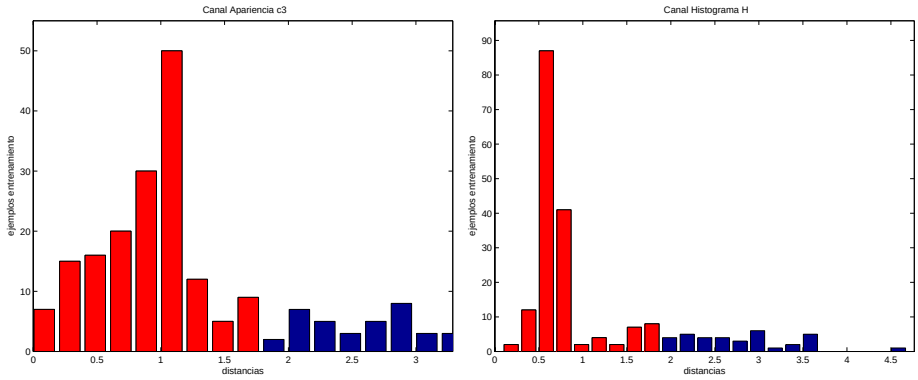


(e) Canal Apariencia c1

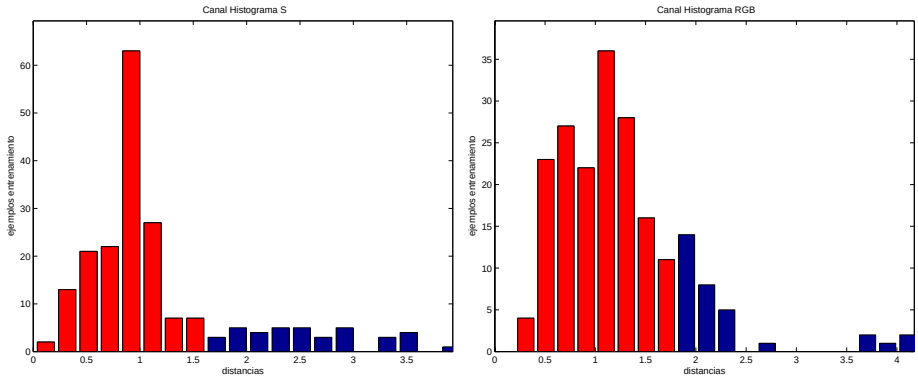


(f) Canal Apariencia c2

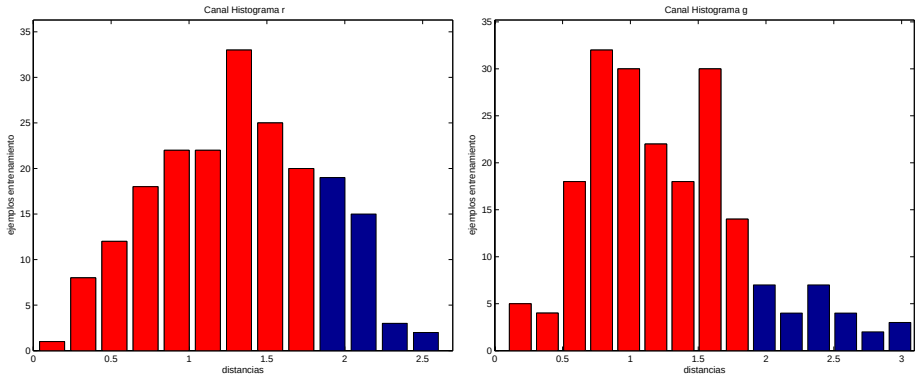
Figura C.9: Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Apariencia b, l1, l2, l3, c1 y c2.



(a) Canal Apariencia c3 (b) Canal Histograma H

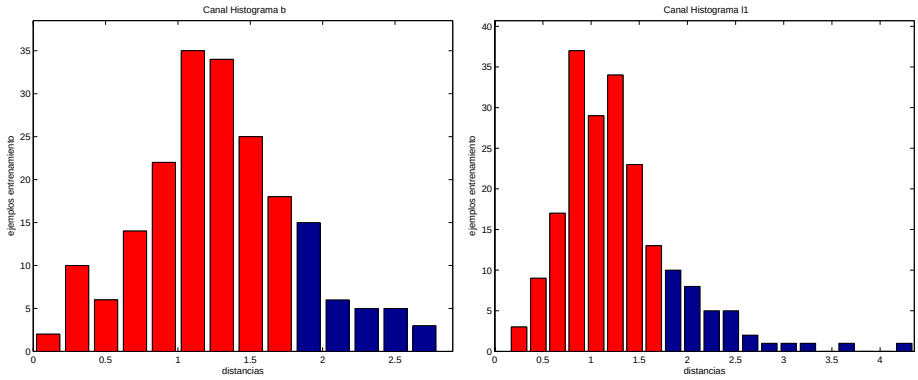


(c) Canal Histograma S (d) Canal Histograma RGB



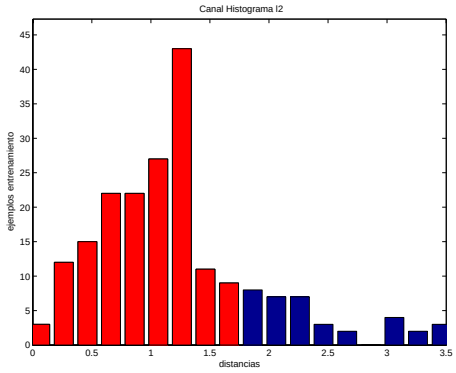
(e) Canal Histograma r (f) Canal Histograma g

Figura C.10: Distribuciones espaciales de los modelos del objeto *la caja amarilla* en el canal de Apariencia c3 y en los canales de Histograma H, S, RGB, r y g.

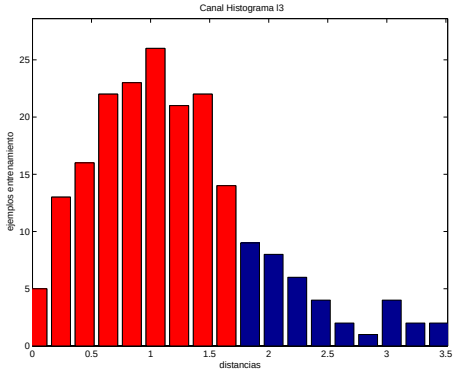


(a) Canal Histograma b

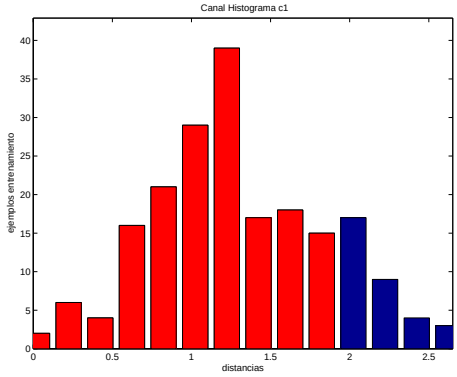
(b) Canal Histograma l1



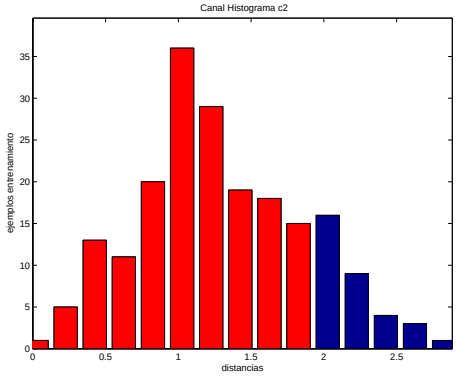
(c) Canal Histograma l2



(d) Canal Histograma l3

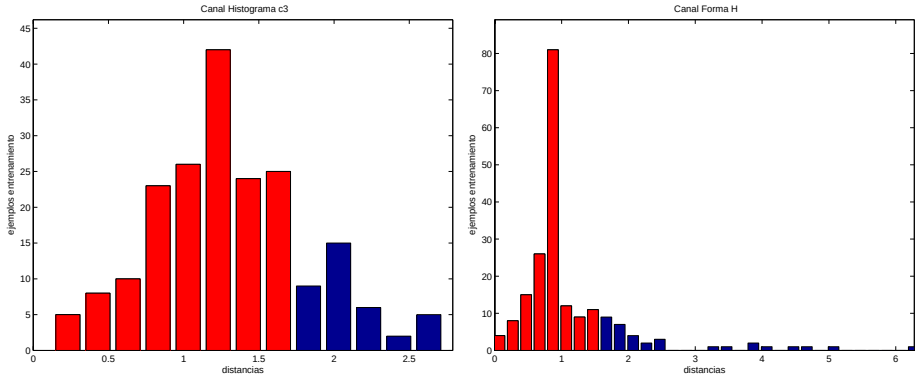


(e) Canal Histograma c1

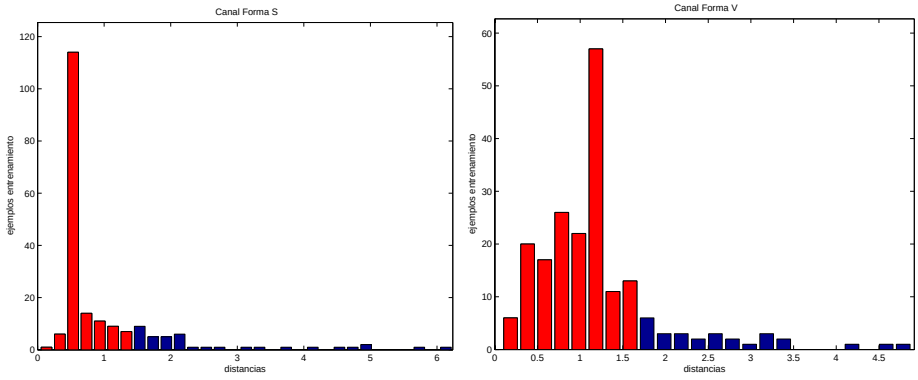


(f) Canal Histograma c2

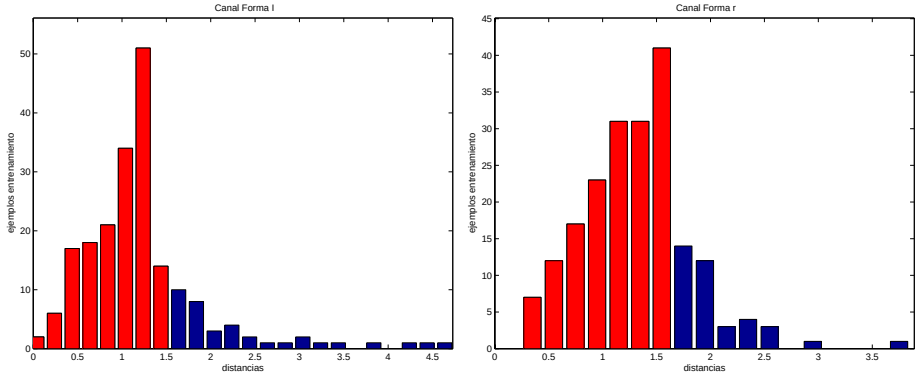
Figura C.11: Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Histograma b, l1, l2, l3, c1 y c2.



(a) Canal Histograma c3 (b) Canal Forma H

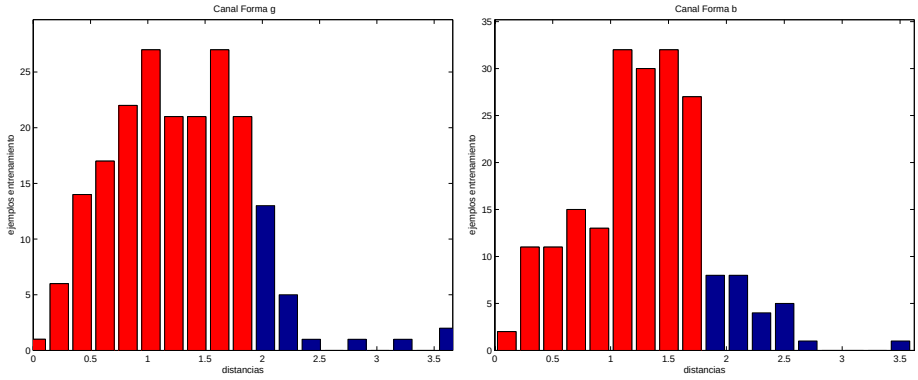


(c) Canal Forma S (d) Canal Forma V



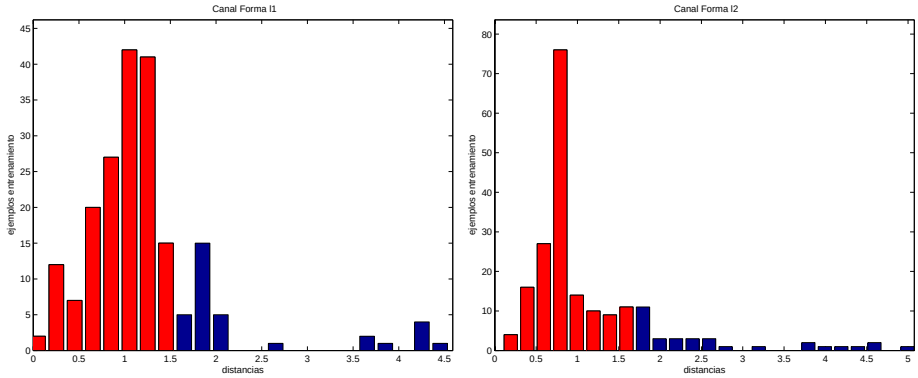
(e) Canal Forma I (f) Canal Forma r

Figura C.12: Distribuciones espaciales de los modelos del objeto *la caja amarilla* en el canal de Histograma c3 y en los canales de Forma H, S, V, I y r.



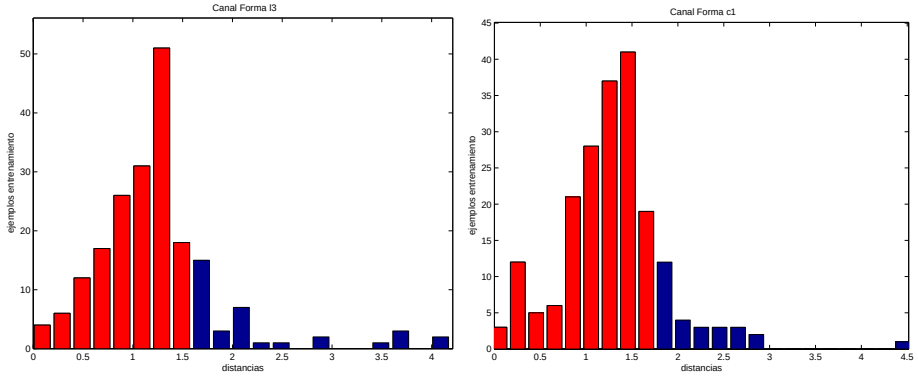
(a) Canal Forma g

(b) Canal Forma b



(c) Canal Forma l1

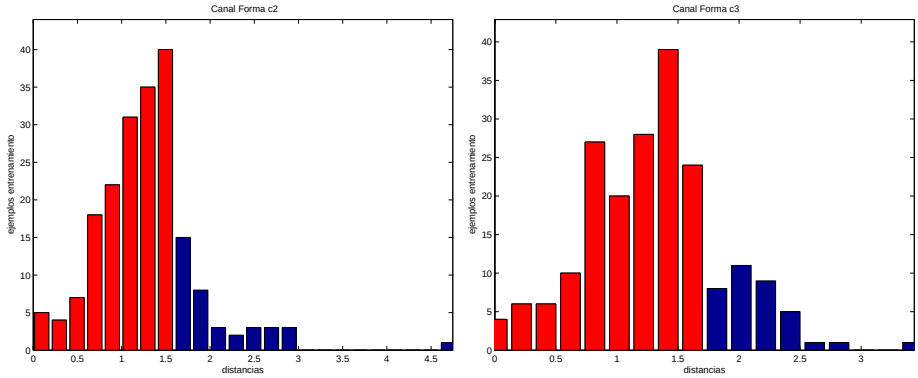
(d) Canal Forma l2



(e) Canal Forma l3

(f) Canal Forma c1

Figura C.13: Distribuciones espaciales de los modelos del objeto *la caja amarilla* en los canales de Forma g, b, l1, l2, l3 y c1.



(a) Canal Forma c2 (b) Canal Forma c3

Figura C.14: Distribuciones espaciales de los mModelos del objeto *la caja amarilla* en los canales de Forma c2 y c3.

Bibliografía

[1] Cognex. http://www.cognex.com/.

[2] Librería de caras CBCL. http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html.

[3] Librería de caras ORL. http://www.uk.research.att.com/facedatabase.html.

[4] Pollux. http://www.pollux.com.br/espa/.

[5] D. Achlioptas. Database-friendly random projections. *ACM Symposium on the Principles of Database Systems*, pages 274–281, 2001.

[6] J. Anderson, A. Pellionisz, and E. Rosenfeld. *Neurocomputing 2: Directions for Research*. MIT Press, 1990.

[7] K. Baek, B.A. Draper, J.R. Beveridge, and K. She. PCA vs. ICA: A comparison on the FERET data set. *Joint Conference on Information Sciences*, Durham, 2002.

[8] R. Barbolla, E. Cerdá, and P. Sanz. *Optimización matemática: teoría, ejemplos y contraejemplos*. Espasa Calpe, 1991.

[9] R.J. Baron. Mechanisms of human facial recognition. *Int. J. Man. Machine Studies*, 15:137–178, 1981.

[10] M.S. Bartlett, H.M. Lades, and T.J. Sejnowski. Independent component representations for face recognition. *SPIE Symposium on Electronic Imaging:Science and Technology; Conference on Human Vision and Electronic Imaging III*, 1998.

[11] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *IEEE Transaction on Neural Networks*, 13:1450–1464, 2002.

[12] M.S. Bartlett and T.J. Sejnowski. Viewpoint invariant face recognition using independent component analysis and attractor networks. *Neural Information Processing Systems - Natural and Synthetic*, 9:817–823, 1997.

[13] W.Mel Bartlett. SEEMORE:combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, (9):777–804, 1997.

[14] P.N. Belhumeor, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[15] A.J Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(1129-1159), 1995.

[16] A.J Bell and T.J. Sejnowski. The independent components of natural scenes are edge filters. *Vision Research*, 37(3327-3338), 1997.

[17] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, New Yersey, 1961.

[18] P. J. Besl. The free-form surface matching problem. *Machine Vision for Three-Dimensional Scenes, Academic Press*, pages 25–71, 1990.

[19] I. Biederman. Recognition by components: a theory of human image understanding. *Psychol. Review*, 94:115–147, 1987.

[20] E. Bienenstock, S. Geman, and D. Potter. Compositionality, MDL priors, and object recognition. *Neural Information Processing Systems*, 9, 1997.

[21] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. *Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, 2001.

[22] H. Bischof and A. Leonardis. Robust recognition of scaled eigenimages through a hierarchical approach. *Proc. IEEE Conf. Computer Vision and Pattern Recognition, Santa Barbara, California*, pages 664–670, 1998.

[23] Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[24] M.J. Black and A.D. Jepson. Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26:63–84, 1998.

[25] A. Blake and M. Isard. *Active Contours*. Springer, 1998.

[26] M. Bressan and J. Vitria. On the selection and classification of independent features. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1312–1317, 2003.

[27] L. Briedman, J.h. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[28] R. Brunelli and T. Poggio. Face recognition: features versus templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[29] R. Campbell and P. Flynn. Eigenshapes for 3D object recognition in range data. *Proc. IEEE. Conf. Computer Vision and Pattern Recognition, Fort Collins, Colorado*, 1999.

[30] O.I. Camps, C.Y. Huang, and T. Kanungo. Hierarchical organization of appearance-based parts and relations for object recognition. *Proc. IEEE. Conf. Computer Vision and Pattern Recognition, Santa Barbara, California*, pages 685–691, 1998.

[31] J. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[32] J.F. Cardoso. Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2655–2658, 1990.

[33] J.F. Cardoso. Infomax and maximu likelihood for source separation. *IEEE Letters on Signal Processing*, 4:112–114, 1997.

[34] J.F. Cardoso and A. Soloumiac. Blind beamforming for non gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[35] H. Cevikalp, M.Ñeamtu, M. Wilkes, and Barkana A. Discriminative common vectors for face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(1):4–13, 2005.

[36] P.A. Chou. Optimal partitioning for classification and regression trees. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(4):340–354, 1991.

[37] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(287-314), 1994.

[38] S. Dasgupta. Experiments with random projection. *Uncertainty in Artificial Intelligence*, (143-151), 2000.

[39] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report 99-006, International Computer Science Institute, Berkeley, California, 1999.

[40] P. Dayan and L. F. Abbott. *Theoretical Neuroscience*. MITT Press, 2001.

[41] G. DeAngelis, I. Ohzawa, and R. Freeman. Receptive-field dynamics in the central visual pathways. *Trends Neuroscience*, 18(10):451–458, 1995.

[42] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[43] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks: Theory and Applications*. Wiley, 1996.

[44] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski. Classifying facial actions. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.

[45] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition in parts? *Advances in Neural Information Processing 16(Proc. NIPS03)*.

[46] C. Dorai and A.K. Jain. COSMOS - a representation schem for 3D free-form objects. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19:1115–1130, 1997.

[47] B. Draper, K. Baek, M.S Bartlett, and R. Beveridge. Recognizing faces with PCA and ICA. *Comput. Vis. Image Underst*, 91(1-2), 2003.

[48] Richard.O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[49] Shimon Edelman. Computational theories of object recognition. *Trends in Cognitive Sciences*, 1:296–304, 1997.

[50] J. Edwards and H. Murase. Appearance matching of occluded objects using coarse-to-fine adaptive masks. *Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pages 533–539, 1997.

[51] K.H. Ekenel and B. Sankur. Multiresolution face recognition. *Image and Vision Computing*, 23(5):469–477, 2005.

[52] H. Etemad and R. Chellapa. Discriminat analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14:1724–1733, 1997.

[53] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[54] Tao Feng, Stan Z.Li, Heung-Yeung Shum, and HongJiang Zhang. Local non-negati e matrix factorization as a visual representation. *Proceedings of the 2nd International Conference on Development and Learning*, 2002.

[55] D.J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[56] R.A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.

[57] P. Földiák and M.P. Young. Sparse coding in the primate cortex. *In Michael A. Arbib editor: The Handbook of Brain Theory and Neural Networks, The MIT Press, Cambridge, Massachusetts*, pages 895–898, 1995.

[58] J. Fortuna, , and D. Capson. Ica for position and pose measurement from images with occlusion. *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '02).*, 4:3604–3607, 2002.

[59] J. Fortuna, , P. Quick, and D. Capson. A comparison of subspace methods for accurate position measurement. *6th IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 16–20, 2004.

[60] J. Fortuna and D. Capson. Ica filters for lighting invariant face recognition. *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 1:334–337, 2004.

[61] J. Fortuna, D. Shuurman, and D. Capson. A comparison of pca and ica for object recognition under varying illumination. *Proc. 16th. International Conference on Pattern Recognition*, 3:11–15, 2002.

[62] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 517 – 522, 2003.

[63] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[64] S. Geman. Minimum description length priors for object recognition. *Challenging the frontiers of knownledge using statistical science*, 1997.

[65] J.M Geusebroek, G.J. Burghouts, and W.M. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

[66] Theo Gevers and W.M. Smeulders. Color based object recognition. *Pattern Recognition*, pages 453–464, 1999.

[67] P. Gochin, M. Colombo, and Dorfman G. Neural ensemble coding in inferior temporal cortex. *J.Neuropsychol.*, (71):2325–2337, 1994.

[68] D. Guillamet. *Statistical Local Appearance Models for Object Recognition*. PhD thesis, Universitat Autònoma de Barcelona, 2004.

[69] D. Guillamet, J. Vitria, and Schiele B. Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters*, 24(14):2447–2454, 2003.

[70] D. Hall, J.L. Crowley, and V. Colin de Verdicre. View invariant object recognition using coloured receptive fields. *Machine GRAPHICS & VISION, Special Issue*, 9(2), 2000.

[71] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[72] S. Haykin. *Neural Networks, A Comprenhensive Foundation*. Prentice Hall International Editions, 1994.

[73] R. Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence:Imitating Life*, pages 43–56, 1994.

[74] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1941.

[75] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[76] C.Y. Huang, O.I. Camps, and T. Kanungo. Object recognition using appearance-based parts and relations. *Proc. IEEE. Conf. Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pages 877–883, 1997.

[77] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[78] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[79] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, New York, 2002.

[80] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[81] A. Hyvärinen and E. Oja. Independent component analysis by general nonlinear hebbian-like learning rules. *Signal Processing*, 64(3):301–313, 1998.

[82] Witten I.H. and E. Frank. *Data Mining*. Morgan Kaufmann Publishers, 2000.

[83] Intel. Open source computel vision library. http://www.intel.com/technology/computing/opencv/.

[84] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[85] A.K. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19, 1997.

[86] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mapping into Hilbert space. *Contemporary Mathematics, American Mathematical Society*, 26:189–206, 1984.

[87] C. Jutten. *Calcul neuromimétique et traitment du signal, analyse en composantes indépendentes*. PhD thesis, INPG, Univ. Grenoble, 1987.

[88] S. Kaski. Dimensionality reduction by random mapping. *Proc. Int. Joint. Conference on Neural Networks*, 1:413–418, 1998.

[89] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Proc. of First International Conference on Computer Vision*, 1:259–269, 1987.

[90] M. Kirby. *Geometric Data Analysis*. Wiley Interscience, 2001.

[91] M. Kirby and L. Sirovich. Application of karhunen-loeve procedure for characterization of human faces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, (12):103–108, 1990.

[92] T. Kohonen. *Self-Organizing Maps*, volume 30. Springer Series in Information Sciences, 1995.

[93] D. Koubaroulis, J. Matas, and Kittler J. Evaluating colour-based object recognition algorithms using the soil-47 database. *Proc. 5th Asian Conference on Computer Vision*, pages 840–845, 2002.

[94] J. Krumm. Eigenfeatures for planar pose measurement of partially occluded objects. *Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Francisco, California*, pages 55–60, 1996.

[95] M. Kurimo. Indexing audio documents by using latent semantic analysis and som. *Kohonen Maps*, pages 363–374, 1999.

[96] D.D. Lee and Seung H.S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 1999.

[97] D.D. Lee and Seung H.S. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562, 2001.

[98] T.W. Lee, M. Girolami, and T.J. Sejnowski. Independent component analysis using and extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11(2):417–441, 1999.

[99] A. Leonardis, H. Bischof, and J. Maver. Multiple eigenspaces. *Pattern Recognition*, 35:2613–2627, 2002.

[100] S.Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized parts-based representations. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1:207–212, 2001.

[101] C. Liu. Enhanced independent component analysis and its application to content based face image retrieval. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(2):1117–1127, 2004.

[102] C. Liu and H. Weschler. Comparative assessment of independent component analysis (ICA) for face recognition. *International Conference on Audio and Video Based Biometric Person Authentification*, Washington D.C., 1999.

[103] C. Liu and H. Weschler. Robust coding schemes for indexing and retrieval from large face databases. *IEEE Trans. on Image Processing*, 9(1):132–137, 2000.

[104] C. Liu and H. Weschler. Independent component analysis of gabor features for face recognition. *IEEE Trans. on Neural Networks*, 14(4):919–928, 2003.

[105] A. Low. *Computer Vision and Image Processing*. McGraw-Hill, New York, 1991.

[106] D.G. Lowe and T.O. Binford. The recovery of three-dimensional structure from image curves. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 7(3):320–326, 1985.

[107] Q.T. Luong. Invariant of a pair of non-coplanar conics in space: Definition, geometric interpretation and computation. Technical report, INRIA, 1989.

[108] D. Marr. *Vision*. W.H. Freeman, San Francisco, CA, 1982.

[109] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three dimensional structure. *Proceedings of the Royal Society of London B*, 200:269–294, 1978.

[110] T. Martiriggiano, M. Leo, T. D'Orazio, and A. Distante. Face recognition by kernel independent component analysis. *Lecture Notes in Artificial Intelligence*, 3533:55–58, 2005.

[111] A.M. Martínez and A.C. Kak. Pca versus lda. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[112] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[113] B. Moghaddam. Principal manifolds and bayesian subspaces for visual recognition. *International Conference on Computer Vision*, Corfu, Greece, 1999.

[114] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):1–10, 2002.

[115] S. Mukherjee and S.K. Nayar. Automatic generation of RBF networks. Technical Report CUCS-001-95, Columbia University, 1995.

[116] J. Mundy, A. Liu, N. Pillow, A. Zisserman, S. Abdallah, S. Utcke, S.Ñayar, and Rothwell C. An experimental comparison of appearance and geometric model based recognition. *Object Representation in Computer Vision II*, pages 247–269, 1996.

[117] H. Murakami and V. Kumar. Efficient calculation of primary images from a set of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (4(5)):511–515, 1982.

[118] H. Murase and S.K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, (14):5–24, 1995.

[119] S.K. Nayar, S.A. Nene, and H. Murase. Real-time 100 object recognition system. *Proceedings of ARPA Image Understanding Workshop, ARPA, Palm Springs*, 1996.

[120] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical report, Technical Report CUCS-006-96, http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html, February 1996.

[121] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library(COIL-20). Technical report, Technical Report CUCS-005-96, February 1996, http://www1.cs.columbia.edu/CAVE/research/softlib/coil-20.html, February 1996.

[122] S.A. Nene and S.K. Nayar. A simple algorithm for nearest-neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:989–1003, 1997.

[123] K.L. Oehler and R.M. Gray. Combining image compression and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):461–473, 1995.

[124] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1043–1048, September 1997.

[125] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13(3):222 – 241, July 1980.

[126] E. Oja. From neural learning to independent components. *Neurocomputing*, 22:187–199, 1998.

[127] B.A. Olshausen and Field D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[128] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

[129] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[130] G. Pajares and J.M. De la Cruz. *Vision por computador*. RA-MA, Madrid, 2001.

[131] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: a probabilistic analysis. *Proc. 17th ACM Symp. on the Principles of Database Systems*, pages 159–168, 1998.

[132] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. *ICCV*, 1998.

[133] C. P. Papageorgiou and T. Poggio. A trainable object detection system: Car detection in static images. *MIT AI Memo No. 180.*, 1999.

[134] L. Parra, C. Spence, P. Sajda, A. Ziehe, and K.R. Müller. Unmixing hyperspectral data. *Advances in Neural Information Processing 12(Proc. NIPS99)*, pages 942–948, 2000.

[135] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[136] V. Perlibakas. Distance measures for pca-based face recognition. *Pattern Recognition Letters*, (25):711–724, 2004.

[137] M. Potter. Short-term conceptual memory for pictures. *J.Exp.Psychol.:Human Learning and Memory*, (2):509–522, 1976.

[138] J.R. Quinlan. *C4.5:Programs for Machine Learning*. Morgan Kaufmann, 1993.

[139] A.Ñ. Rajagopalan, P. Burlina, and R. Chellappa. Higher order statistical learning for vehicle detection in images. *ICCV*, pages 1204–, 1999.

[140] M. Rajapakse and L. Wyse. Face recognition with non negative matrix factorization. *IEEE Visual Communications and Image Processing, Lugano*, pages 1838–1847, 2003.

[141] M. Rajapakse and L. Wyse. Nmf vs ica for face recognition. *3rd International Symposium on Image and Signal Processing and Analysis, Roma*, pages 605–611, 2003.

[142] R. Rao. Dynamic appearance-based recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pages 540–546, 1997.

[143] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20:23–38, 1998.

[144] H.S. Sahambi and K. Khorasani. A neural-network appearance-based-3d object recognition using independent component analysis. *IEEE Trans. on Neural Networks*, 14(1):138–149, 2003.

[145] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.

[146] S.A. Shafer. Using color to separate reflection components. *COLOR Res. Appl.*, 10(4):210–218, 1985.

[147] SigCAM. Human face detection and license plate detection. http://www.signalogic.com.

[148] E.P. Simoncelli and B.A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1215, 2001.

[149] D. Socolinsky and A. Selinger. A comparative analysis of face recognition perfomance with visible and thermal infrared imagery. *International Conference on Pattern Recognition*, Quebec, 2002.

[150] C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

[151] B. Stankiewicz and J. Hummel. Metricat: a representation for basic and subordinate-level classification. *Proc. of 18th Annual Conf. on the Cognitive Science Society*, pages 254–259, 1996.

[152] F. Stein and G. Medioni. Structural indexing:efficient 3-d object recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14:125–145, 1992.

[153] K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[154] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[155] D.L. Swets and J.J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

[156] S. Thorpe. Localized versus distributed representations. *In Michael A. Arbib editor: The Handbook of Brain Theory and Neural Networks, The MIT Press, Cambridge, Massachusetts*, pages 549–552, 1995.

[157] M.A. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[158] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2002.

[159] M.A. Vicente. Reconocimiento de objetos 3D mediante invariantes proyectivos. Master's thesis, EPSE, Universidad Miguel Hernández, 2001.

[160] M.A. Vicente, C. Fernández, Reinoso O., and L. Payá. 3D object recognition from appearance: PCA versus ICA approaches. *Lecture Notes in Computer Science*, 3211:547–555, 2004.

[161] M.A. Vicente, C. Fernández, R. Puerto, and A. Gil. Extracción de características para reconocimiento visual. *Actas de las XXV Jornadas de Automática*, pages 1–8, 2004.

[162] M.A. Vicente, C. Fernández, and O. Reinoso. Identificación de objetos en entornos no estructurados utilizando técnicas basadas en la apariencia. *Seguimiento e Identificación de Objetos*, pages 29–38, 2004.

[163] M.A. Vicente, C. Fernández, and O. Reinoso. Object discrimination by appearance-based visual learning. *Proceedings of the Learning 04 International Conference*, pages 129–134, 2004.

[164] M.A. Vicente, C. Fernández, and O. Reinoso. Recognizing objects in non-controlled backgrounds by an appearance two-step approach. *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 2565–2570, 2004.

[165] M.A. Vicente, C. Fernández, and O. Reinoso. Visual detection of free-form objects at multiple scales using appearance-based local features. *Machine Vision and Applications*, Pendiente de aceptación.

[166] M.A. Vicente, A. Gil, C. Fernández, L.M. Jiménez, and R. Ñeco. Appearance-based recognition with varying patterns. *IADAT International Conference on Multimedia, Image Processing and Computer Vision*, pages 188–192, 2005.

[167] M.A. Vicente, A. Gil, C. Fernández, L.M. Jiménez, and R. Ñeco. Object recognition with random projection. *IADAT Journal of Advanced Technology on Digital Image Processing*, pendiente de publicación.

[168] M.A. Vicente, P. Gil, O. Reinoso, and F. Torres. Object recognition by means of projective invariants considering corner-points. *Journal of WSCG*, pages 129–136, 2002.

[169] M.A. Vicente, P. Hoyer, and A. Hyvarinen. ICA vs. PCA as preprocessing for appearance-based object recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, En proceso de revisión.

[170] M.A. Vicente, O. Reinoso, N. García, J.M. Sabater, C. Fernández, and L. Jiménez. Experiments with COIL database using PCA-based object recognition techniques. *Proceedings of the 3rd International Symposium on Robotics and Automation*, pages 380–385, 2002.

[171] M.A. Vicente, O. Reinoso, C. Pérez, and C. Fernández. Recognition and location of real objects using eigenimages and a neural network classifier. *Proceedings of SPIE on Visual Communications and Image Processing*, 5150:385–392, 2003.

[172] M.A. Vicente, O. Reinoso, C. Pérez, J.M. Sabater, and J.M. Azorín. Reconocimiento de objetos 3D mediante análisis PCA. *Actas de las XXIII Jornadas de Automática*, pages 1–8, 2002.

[173] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[174] A. Webb. *Statistical Pattern Recognition*. Wiley, 2004.

[175] L. Wittgenstein. *Philosophical Investigations*. Blackwell, London, 1973.

[176] Q.B. Xie, C.A. Laszlo, and R.K. Ward. Vector quantization technique for non-parametric classifier design. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(12):1326–1330, 1993.

[177] M.H. Yang, D. Roth, and N. Ahuja. A snow-based face detector. *Advances in Neural Information Processing Systems 12*, pages 851–861, 2000.

[178] P.C. Yuen and J.H. Lai. Independent component analysis of face images. *IEEE Workshop on Biologically Motivated Computer Vision*, Seoul, 2000.

[179] M. Zerroug and G. Medioni. The challenge of generic object recognition. *Object Representation in Computer Vision.Proc. NSF-ARPA Workshop*, pages 217–232, 1994.