3D Object Recognition from Appearance: PCA vs. ICA Approaches

M. Asunción Vicente[1], Cesar Fernández[1], Oscar Reinoso[1], Luis Payá[1]

[1] Industrial Systems Engineering Department,
Miguel Hernandez University, Avda. de la Universidad s/n,
03202 Elche, Spain
{suni, c.fernandez, o.reinoso, lpaya} @umh.es
http://lorca.umh.es

Abstract. Two feature extraction techniques (PCA/ICA) for recognition of 3D objects from appearance are compared with respect to different recognition approaches (universal/object subspaces). A class separation ratio is defined, and several recognition experiments are performed using the COIL-100 database. The results show that both techniques produce similar recognition rates when universal subspaces are used; but, when object subspaces are used, ICA representation greatly outperforms the earlier PCA technique due to its ability to separate classes.

1 Introduction and Motivation

Object recognition is a fundamental ability of any visual system. Recognizing real three-dimensional objects in controlled backgrounds or scenes is quite easy provided an adequate 3D visual model of the object is available. However, non-controlled backgrounds make it almost impossible to apply such model-based techniques because in these situations it is very difficult to segment the target object from the scene. Appearance-based recognition approaches are a powerful alternative to model-based techniques when it is difficult to obtain geometrical models of the objects [1] and when the images have non-controlled backgrounds [12] [9]. First appearance-based systems found in the literature used principal component analysis (PCA) as a feature extraction technique to reduce the dimensionality of the object classes or models [7][10], while recently the use of the independent component analysis (ICA) for feature extraction is preferred by some authors [2]. The performances of both techniques have not been compared in terms of their applicability to object recognition and sometimes contradictory conclusions have been drawn. The aim of this paper is to clarify the advantages and drawbacks of both techniques as feature extractors in object recognition systems.

In [7] Murase addresses the problem of automatically learning object models for recognition and pose estimation. PCA is used and the objects are represented in two different *eigenspaces*: the *universal eigenspace*, computed using an image set of all objects of interest, and the *object eigenspaces*, computed using only images of an object. The universal eigenspace is best suited for discriminating between objects, whereas the object eigenspace is better for pose estimation. In this paper the structure of the PCA-based Murase recognition system is used to perform the comparison of ICA and PCA approaches with the COIL100 library [8]. The results presented throughout the paper show how ICA clearly outperforms PCA when using object subspaces. The reason can be found in its class separation capability, as stated by Bressan [3].

2 Recognition of 3-D Objects from Appearance

The appearance of a 3D object in a 2D image depends on its shape, its colour, its pose in the global scene, its reflectance properties and the sensor and illumination characteristics. An object image may be considered as a vector of pixels where the value of each entry in the vector is the greyscale (or colour) value of the corresponding pixel. For example, a NxN image may be unwrapped and treated as a vector of length N^2. The image is said to sit in N-dimensional space, which is considered to be the original space of the image. In appearance-based systems the whole object image is projected to a lower dimensional space using different techniques or subspaces, the most frequently used is the subspace created by the eigenvectors of the covariance matrix of the training images (or PCA) [7] [10], another common subspace is the one created by the basis vectors obtained using Linear Discriminant Analysis or the subspaces computed by Independent Component Analysis [6].

2.1 Constructing the *Object Subspaces*

Two different representations are used: namely the *universal subspace* and the *object subspaces* following Murase nomenclature [7]. Each object of interest can be represented or modelled using an *object subspace*. This subspace is constructed from a set of training images belonging to different views of the same object. The set of M training images, $\vec{x}_m$, is defined by G_j:

$$G_j = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_M\}. \quad (1)$$

and there are as many sets as different objects of interest. With $\vec{x}_m$, we denote a vector image that groups the three color image matrices. Each image, $\vec{x}_m$, from the training set is filtered by the feature channel selected to model the object:

$$\vec{f}_m = F_{channel}(\vec{x}_m). \quad (2)$$

Typical feature channels are: the image in a modified colour space, the image histogram, the border image, or special areas from the image grouped in feature windows (local image features) [2][9]. Once the feature channel is selected, the new set of M training feature vectors, G_j' is defined:

$$G_j' = \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_M\}. \quad (3)$$

The object subspace, W, is computed from G_j' for each object *j*, extracting the principal components (PC) or the independent components (IC). After that, all the training image features are projected in that subspace creating an object representation, named as *manifold.* That projection can be represented as:

$$\vec{p}_m = W \cdot \vec{f}_m. \quad (4)$$

where W is the matrix containing the PC's or the IC's from each object.

2.2 Constructing the *Universal Subspace*

The universal subspace is constructed from a set of training image features belonging to different objects. The set of J training images is defined by U:

$$U = \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_J\}. \quad (5)$$

The universal subspace, W, is also computed by PC or IC extraction. But in this case there is only one subspace for all the objects. So, the model of each object is composed of its views projected in the universal subspace.

3. Selection of Features Using Linear Transformations

3.1 PCA Approach

PCA subspace or eigenspace is computed by finding the eigenvectors of the covariance matrix created from the set of training vectors [4]. The eigenvectors corresponding to non-zero eigenvalues of the covariance matrix represent an orthonormal basis that projects the original vectors (of length N^2) in the M-dimensional space ($M \ll N^2$).

So in PCA approach, W is the matrix containing the eigenvectors. In this work the *snapshot* method [10] has been used to compute the eigenspace in order to avoid the high dimensionality of the original covariance matrix. PCA enables us to create and use a reduced set of variables. A reduced set (the classes obtained from the training images) is much easier to analyze and interpret than the original variables (the training images themselves).

3.2 ICA Representation

The ICA of an N^2 dimensional random vector is a linear transform that minimizes the statistical dependence between its components. This analysis has a great number of applications such as data analysis and compression, blind source separation, blind deconvolution, denoising, etc.

If the random vector we wish to represent through ICA has no noise and is zero-centered, the ICA model can be expressed as:

$$\vec{x} = A \cdot \vec{s}\,. \tag{6}$$

where $\vec{x}$ is the random vector representing our data, $\vec{s}$ is the random vector of independent components with dimension $M \le N^2$, and A is the mixture matrix. The pseudoinverse of A, represented by W, is called the projection matrix and it provides an alternative representation of the ICA model:

$$W \cdot \vec{x} = \vec{s}\,. \tag{7}$$

Various objective functions have been proposed for the estimation of the projection matrix such as nongaussianity, likelihood, mutual information, and tensorial methods [6]. In this paper the FastICA [5] method is used. FastICA estimates the whole decomposition by minimizing mutual information, and estimates the individual independent components as projection pursuit directions. FastICA uses PCA as a pre-processing step for data whitening.

4 Experimental Results with the COIL-100

The experiments are performed with object images from the COIL-100 database [8]. The selected feature channel to carry out the comparison has been the *I* component (*HIS* colour space).

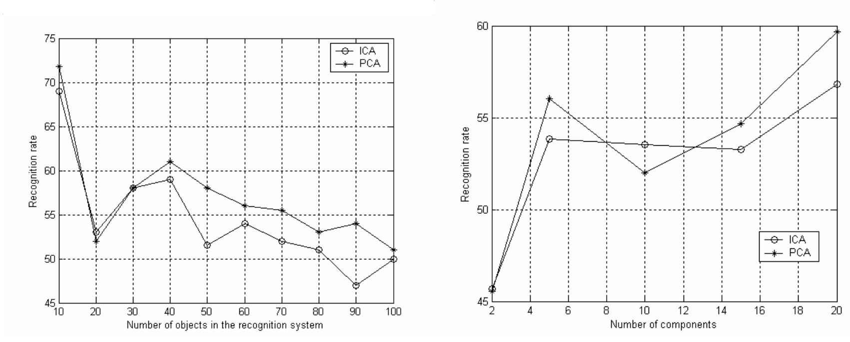
4.1 Comparing *Universal Subspaces*

In order to compare the performance of PCA and ICA universal subspaces were computed from the main image of each object in the database (10 to 100 object sets were used in different experiments). The rest of the images were used as test examples and the classifier selected was the nearest neighbour (k-NN with k=1 and L2-norm as distance measure). In Fig. 1 the experimental results from both feature selectors are shown: both techniques performed almost equally. Fig. 1a shows the recognition rates obtained using both ICA and PCA when only 10 components are used to represent each subspace. As the number of different objects increases, there is a reduction in classification accuracy with both systems. In Fig. 1b the influence of the dimensionality of the subspace is shown: all experiments are carried out with 20 different objects, and the number of components varies from 2 to 20. As expected, the recognition rate grows as the dimensionality of the subspace increases. Experiments performed with bigger object sets gave similar results.

The universal subspaces perform well with symmetrical objects, as all the views of an object are very similar; but it produces disastrous results with quite non-symmetrical objects. This is shown in Fig. 2: the symmetrical objects, (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), give a recognition rate of 100% in both types of subspaces whereas non-symmetrical objects are hardly recognized. We can also notice than ICA works only slightly worse than PCA as there are 26 symmetrical objects in PCA's case and just 21 in ICA's (5, 18, 24, 25, 26, 30, 32, 34, 35, 47, 50, 56, 58, 61, 70, 72, 83, 86, 87, 94, 95). These similar results were expected as ICA representation is based on a dimensionality reduction obtained by using PCA in the pre-processing step of the FastICA algorithm and the further statistical independence forced by ICA does not necessarily improve the results.

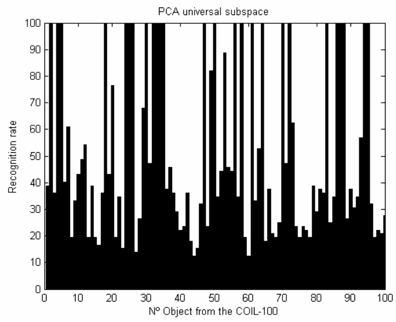
As a conclusion, universal subspaces with a distance measure can be used to recognize objects when they are quite symmetrical or they always show the same view to the camera. Universal subspaces are mostly used in face recognition systems with the processed frontal face (*mugshots*) as the training images.

If a more complex classifier is used instead of a distance measure, the recognition rates can be improved even with non-symmetrical objects; on the other hand, this requires the use of more training views in the learning step [12].

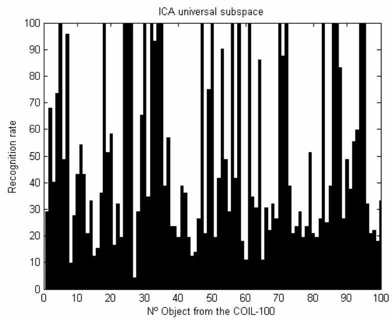


(a) (b)

Fig.1. Comparing universal subspaces: Fig 1.a shows ICA and PCA recognition rates using just 10 dimensions, and Fig 1.b is referred to the influence of the subspace dimensionality.



(a)



(b)

Fig. 2a. Symmetrical objects in PCA subspace. **Fig. 2b.** Symmetrical objects in ICA subspace.

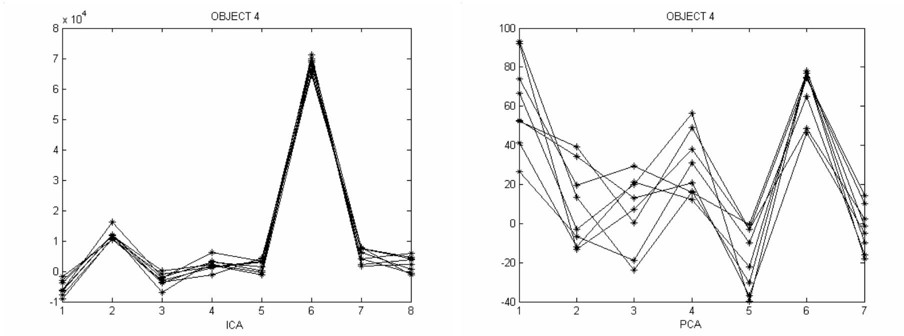
4.2 Comparing *Object Subspaces*

Object subspaces generated with PCA were used by Murase in [7] in order to find out the pose of the object and not to identify it, but they are also useful for recognition as described in [11]. Even more, if ICA is used instead of PCA, this technique greatly outperforms the universal subspace method as the ICA manifolds of each class are more apart from each other. The reason can be found in the close relationship existing between sparse coding and ICA: sparse coding is a coding of the data such that only a few components of the code will be significantly active (nonzero) In Fig. 3a it is possible to observe the high sparsity in the ICA manifolds (each object fires a certain component, while the remaining ones are kept close to zero). PCA manifolds do not represent a sparse coding as each component is not associated to a certain object; the components are instead ordered in decreasing values of their variance in the training data (Fig. 3b).

In order to compare the two approaches a class separation ratio has been defined. This ratio represents the degree of separation of a certain class, and it can be expressed as follows:

$$\alpha_j = \log\left[\frac{\max(d_j)}{\min(D_j)}\right]. \qquad (\mathbf{8})$$

where α_j is the class separation ratio for class j, d_j is the L2 norm between two projection vectors belonging to manifold j (distances between elements of the same class) and D_j is the L2 norm between two projection vectors, only one of them belonging to manifold j (distances between elements of different classes).



(a) (b)

Fig. 3a. Values of the ICA projections over an 8 component subspace (object 4). The sparsity of the code is evident: all but one of the components are close to zero. **Fig. 3b.** Values of the PCA projections over a 7 component subspace (same object). It can be seen that the components are ordered in decreasing values of variance.

High α_j values represent easily separable classes, and particularly, when α_j is greater that zero, the classes can be trivially separated by any simple classifier. The class separation ratio for ICA and PCA object subspaces for the whole library is shown in Fig. 4a : most of the ica-classes from the COIL-100 are above or close to the threshold $\alpha_j = 0$. However the pca-classes are in almost all objects below this threshold, meaning that they are not so easily separable.

This fact is also shown in Fig.5a and 5b., where each approach is tested with a different number of training examples (from 2 to 24). It can be seen that PCA class separation ratio falls below the $\alpha_j = 0$ threshold even with 24 training examples; whereas the ICA class separation ratio is always higher than the PCA one and even more, only 4 components are required to trivially separate some of the objects. Several classification experiments have been performed using a linear perceptron, a k-NN rule and a naive Bayes classifier, and the recognition rates obtained confirm the results: with just a few components (8 o more) ICA obtains recognition rates close to 100%, whereas PCA does not reach a 100% classification rate even with 24 components.

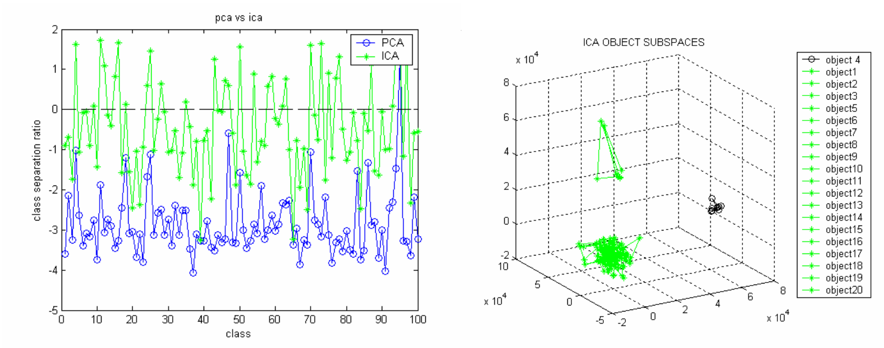
5 Conclusions and Future Work

Two main conclusions are drawn from this work:

- The universal subspace technique with a distance based classifier is only applicable to symmetrical objects or objects that always present the same view to the camera. In this scenario ICA does not improve PCA results.
- The object subspace technique benefits from using ICA instead of PCA. ICA produces a reduced model of the object due to the sparse coding of the CI's;

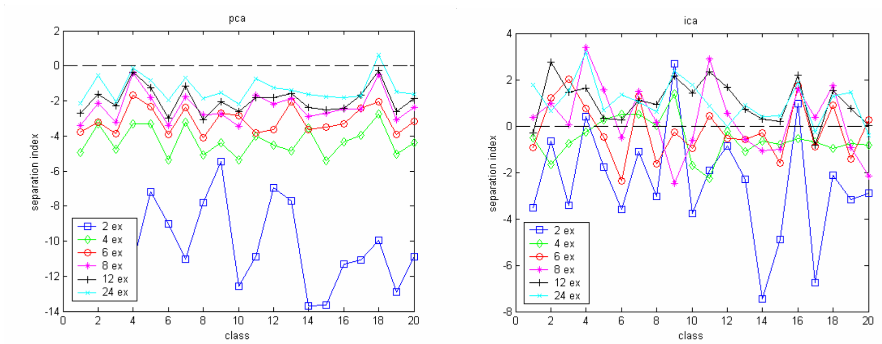
even more, class separation is much more relevant even with only a few components.

An ICA-based object subspace technique allows extremely compact representations of the objects. This fact makes this technique suitable for the combination of different feature channels in order to improve the recognition capability of the visual system. Future works will consider simultaneously color, shape and texture information combined to get an almost invariant representation of the object, useful for object recognition in non-controlled backgrounds.



(a) (b)

Fig. 4a. Class separation ratio for ICA and PCA object subspaces for the whole library. The subspaces are computed using just 8 training images. **Fig. 4b.** In this figure the reduced manifolds of the 20 first objects from COIL-100 are displayed, the three mostly fired components have been selected for a 3D representation .



(a) (b)

Fig. 5. Class separation ratio for PCA(a) and ICA(b) object subspaces for the 20 first objects from COIL-100 using different dimensionalities of the manifolds (from 2 to 24 components). Class separation increases with the dimensionality (larger training sets) in both cases, but ICA clearly outperforms PCA even with a lower number of components.

References

1. Biederman, I: Visual object recognition. In: S. F. Kosslyn and D. N. Osherson (Eds.). An Invitation to Cognitive Science, 2nd edition, Volume 2., Visual Cognition. MIT Press. Chapter 4, pp. 121-165. 1995.
2. Bressan M., Guillamet D. and Vitrià J.: Using an ICA representation of local color histograms for object recognition. Pattern Recognition 36(3): 691-701 (2003)
3. Bressan M. and Vitrià J.: On the Selection and Classification of Independent Features, IEEE Transactions on Pattern Analysis and Machine Intelligence,vol. 25 n.10, pp. 1312- 1317, 2003.
4. Duda Richard.O., Hart Peter E. and Stork David G., Pattern Classification, Wiley-Interscience, 2001
5. FastICA software, http://www.cis.hut.fi/projects/ica/fastica/
6. Hyvärinen A., Karhunen J., and Oja E.: "Independent Component Analysis", Wiley-Interscience, 2002.
7. Murase H. and Nayar S.: "Visual learning and recognition of 3D objects from appearance", International Journal of Computer Vision, no.14, 1995, pages 5-24.
8. Nene S. A., Nayar S. K. and Murase H.: "Columbia Object Image Library (COIL-100)", Technical Report CUCS-006-96, 1996, http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html.
9. Ohba K. and Ikeuchi K., Detectability, Uniqueness, and Reliability of EigenWindows for Stable Verification of Partially Occluded Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19 n. 19, pp. 1043-1048, 1997.
10. Turk M. and Pentland A., Eigenfaces for Recognition, Journal of Cognitive Neuroscience, vol. 3, no. 1, pp.71-86,1991.
11. Vicente M.A., Reinoso O., García N., Sabater J.M., Fernández C., Jiménez L.: Experiments with COIL database using PCA-based Object Recognition Techniques, ISRA'2002 3rd International Symposium on Robotics and Automation, 2002.
12. Vicente M.A. , Reinoso O. , Fernández C.: "Recognition and location of real objects using eigenimages and a neural network classifier", Proceedings of SPIE: Visual Communications and Image Processing, vol. 5150, pp. 385-392, 2003.