Appearance Based 3D Object Recognition Using IPCA-ICA

V.K Ananthashayana and Asha.V

Image Processing and Computer Vision Lab
Department of Computer Science and Engineering
M.S.Ramaiah Institute of Technology
Bangalore, India

Commission Ⅰ, ThS-2

KEY WORDS: Principal component analysis (PCA), Independent component analysis (ICA), principal non-Gaussian directions, blind source separation (BSS), Appearance-based object recognition, Incremental principal component analysis (IPCA_PCA).

ABSTRACT:

Fast incremental non Gaussian directional analysis (IPCA-ICA) is proposed as a linear technique for recognition [1]. The basic idea is to compute the principal components as sequence of image vectors incrementally, without estimating the covariance matrix and at the same time transforming these principal components to the independent directions that maximize the non–Gaussianity of the source. In this paper we utilize the IPCA-ICA technique for 3D Object recognition by employing its neural network architecture. We illustrate the potential of IPCA-ICA on a database of 1440 images of 20 different objects captured by CCD camera. The excellent recognition rates achieved in all the performed experiments indicates that the present method is well suited for appearance based 3D object recognition.

1. INTRODUCTION

A human activity relies heavily on the classification or identification of a large variety of visual objects. In Computer vision, the recognition system typically consists of sensors, and model database in which all the object representations and decision making abilities are saved. Recently, object recognition has been found in a great range of applications like Surveillance, Robot vision, Medical Imaging etc. For the view-based recognition, the representation takes into account the appearance of the object. This requires three-dimensional object recognition (3DOR), for which the pose of the object is of main consideration. The objective of the 3DOR algorithm is not only to recognize the object precisely but also to identify its pose as viewed. Then a recognition algorithm tries to find the best matched object. A schematic of a typical object recognition system is shown in Fig 1.

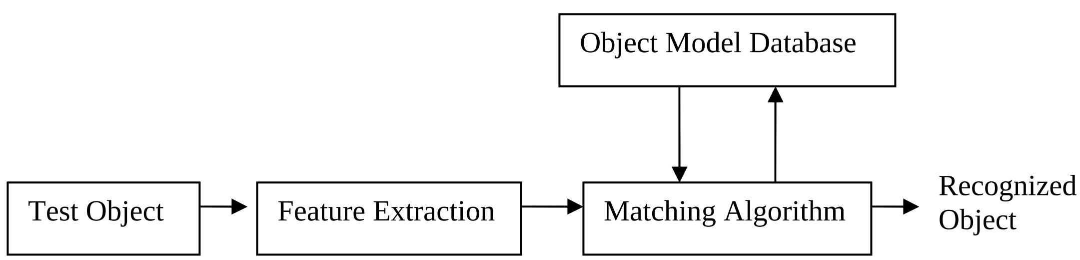


Fig 1. Block Diagram of Typical Object Recognition System

Fig 1 represents the typical object recognition system. In this system, its geometric features like edges, lines, curves and vertices are called geometric features are extracted, since the edges contain more information. In contrast to the geometric features, appearance of an object is the combined effect of its shape, reflectance properties, pose and the illumination. The approaches that take explicitly these factors into account for use in object recognition have been categorized as appearance-based object recognition methods. The main idea in the appearance-based approach is to represent the images in terms of their projection onto a low-dimensional space is called eigenspace. The projection of the images onto this space is called the eigenimages. One popular method to obtain the low dimensional space is called the PCA [3], and other methods like Kernal PCA and independent component analysis (ICA)[2] are used in 3D object recognition. In these methods, the images of the objects are taken from various pose angles, and their compressed features are saved in a database. The test images are taken in similar conditions and the features extracted from them are matched against the database to determine the identity of the object along with its pose angle.

The traditional PCA algorithm [3] computes eigenvectors and eigenvalues for a sample covariance matrix derived from a well-known image data matrix, by solving an eigenvalue system problem. The incremental principal component analysis (IPCA) which is the incremental version of the principal component analysis. The independent component analysis (ICA) [4] is used to separate independent components from the set of unknown mixtures. It is known that there is a correlation or dependency between different objects, the set of objects is represented as a data matrix X. The correlation between the rows of the matrix X can be represented as the mixing matrix A. The independent basis objects are represented as rows of source matrix S. The ICA algorithm extracts these independent objects from the set of dependent ones using (1).ICA is much related to the method called the BSS, where a correlated source is separated into uncorrelated source without prior knowledge about the correlation between the elements of the source. When the dimension of the image is high, both the computation and storage complexity grow dramatically. Thus the idea of using the real time process becomes very efficient in order to compute the principal independent components for observations (objects). Each eigenvector or principal component will be updated using FastICA algorithm, to a non-Gaussian component. Here random vector is said to be non-Gaussian if its distribution is not a Gaussian distribution .In (1) if the source matrix S contains Gaussian uncorrelated elements in the mixed matrix X will also be Gaussian but correlated elements.

The most common ICA algorithm FastICA method does not have a solution if the random variables to estimate, are Gaussian random variables. This is due to the fact that the joint distribution of the elements of X will be completely symmetric and doesn't give any special information about the columns of A. In this paper, S is always a non- Gaussian vector.

$$X=AS \qquad (1)$$

In this paper we have applied the IPCA-ICA to 3D object recognition. The combined method of IPCA and ICA. To the best of our knowledge IPCA-ICA has not been applied to appearance-based **3D** object recognition and pose estimation. The contributions of this paper are in the application of the IPCA - ICA representation based on the work for 3D object recognition, as well as investigating and determining whether IPCA-ICA would always outperform the PCA and ICA in the appearance-based 3D object recognition task.

2. METHODOLOGY

The object recognition can be done by projecting an input image onto the block diagram(Fig 2)and comparing the resulting coordinates with those of the training images in order to find the nearest appropriate image. The database consists of n images and a set of k non-Gaussian vectors. This algorithm takes input image finds the non-Gaussian vector (eigenvector) which is passed as input to the ICA algorithm. The non-Gaussian components will be updated using the updating rule (3) from the previous component values in a recursive way. While IPCA returns the estimated eigenvectors as a matrix that represents subspaces of data and the corresponding eigenvalues as a row vector, FastICA searches for the independent directions as in eq(3) where the projections of the input data vectors will maximize the non-Gaussianity

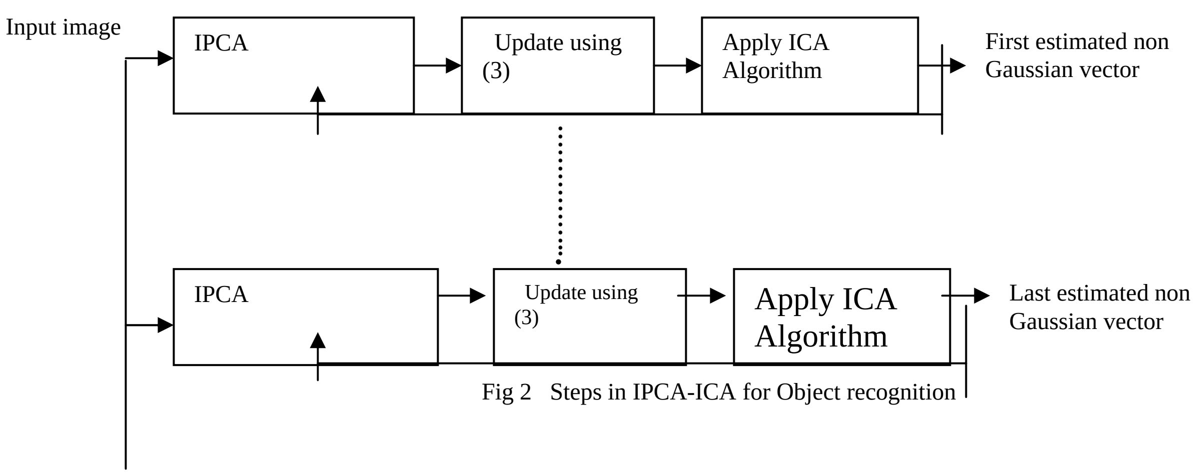


Fig 2 Steps in IPCA-ICA for Object recognition

The object recognition can be done by projecting the input test image onto this basis and comparing the resulting coordinates with those of the training images in order to find the nearest appropriate image. The data consists of n images and a set of k non-Gaussian vectors are given. Initially, all the non-Gaussian vectors are chosen to describe an orthonormal basis. In each step, all those vectors will be updated using an IPCA updating rule(3). Then, each estimated non-Gaussian vector will be an input for the ICA function in order to extract the corresponding non-Gaussian vector from it (Fig. 2).

Mathematically,By definition, an eigenvector x with a corresponding eigenvalue λ of a covariance matrix C satisfies

$$\lambda.x = C.x \qquad (2)$$

By replacing C with the sample covariance matrix $\sum_{i=1}^{n} u(i).u^T(i)$ and using v= $\lambda.x$ we will get the nth eigenvector i.e. *v(n)* for the n images of the database.

Then, this vector will be the initial direction in the FastICA algorithm.

$$w=v(1) \qquad (3)$$

v(1) is the first principal component.

The FastICA[16] algorithms will repeat until convergence the following rule:

$$Wnew=E[v(1).g(w^T.v(1))]-E[g'(w^{T.}v(1))].w \qquad (4)$$

Where g'(x) is the derivative of the function g(x) (6). It should be noted that this algorithm uses an approximation of negentropy in order to assure the non-Gaussianity of the independent vectors. Before starting the calculation of negentropy, a non-quadratic function G should be chosen, for example,

$$G(u)=-exp(-u^2/2 \qquad (5)$$

And its derivative:

$$g(u)=u.exp(-u^2/2) \qquad (6)$$

In general, the corresponding non-Gaussian vector w, for the estimated eigenvector v(1), will be estimated using the following repeated rule:

$$Wnew=E[v(1).g(w^T.v(1))]-E[g'(w^{T.}v(1))].w \qquad (7)$$

The previous discussion only estimates the first non-Gaussian vector. One way to compute the other higher order vectors is to start with a set of orthonormalized vectors, update them using the suggested iteration step and recover the orthogonality. Further, the non-Gaussian vectors should be orthogonal to each other in order to ensure the independency. So, it helps to generate "observations" only in a complementary space for the computation of the higher order eigenvectors. After convergence, the non-Gaussian vector will also be enforced to be orthogonal, since they are estimated in complementary spaces. As a result, all the estimated vectors w_k will be: Non-Gaussian according to the learning rule in the algorithm. . Independent according to the complementary spaces introduced in the algorithm.

The nearest neighbor algorithm is used to evaluate the object recognition technique. Each Object Database is truncated into two sets. The training set that contains images used to calculate the independent non- Gaussian vectors and come up with the appropriate basis and, the test set that contains images to be tested by the Object recognition algorithm in order to evaluate the performance of the proposed method. The whole set of training images (rows in the image data matrix) are projected into the basis found in order to calculate the coordinates of each image with respect to the basis $v_{train.}$ Each new testing image v_{test} is compared to whole set of training images v_{train} in order to come up with nearest one that corresponds to the maximum k in (8).

$$[k] = nearest\_match\ (v_{test.,}\ v_{train.}) \qquad (8)$$

k^{th} image gives the index of the object that is recognized from the database.

3. RESULTS

Object recognition and pose estimation experiments were performed by using Matlab7.1. The object set is COIL-20 (Columbia Object Image Library) database [14]. The images are stored at every 5^0 of pose angle, from 0^0 to 360^0. Hence 72 images of each object, and 1440 total number of images. The size of the images is rescaled to 64x64.The 0^0 pose angle views are shown in Fig.3 the maximum pixel value is 255.

Fig 3 COIL database of 20 objects

To construct the non Gaussian space of the object, a few of the images were chosen as the training images. The representations of images make a manifold with the pose angle as the parameter

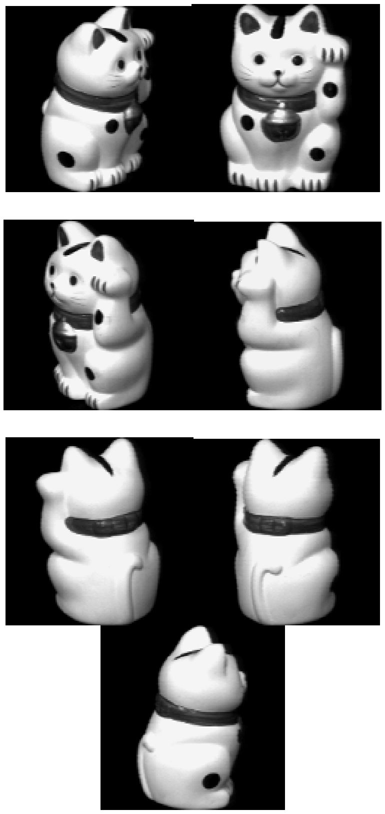


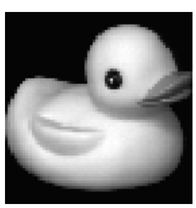
Fig 4 Training images of the fourth objects sampled by every 50^0

in the high-dimensional space and the image manifold is sampled at regular intervals of pose angle to make the training images. In the first experiment, images separated by 50^0 in pose angle were chosen to construct the representative non Gaussians space of the images. That is to say, there are 7 training images for each object, making a total of 140 training images. The training views of the first object are shown in Fig 4

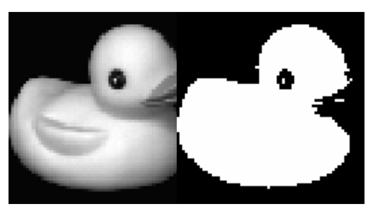
3.1. Results with Pose Angle Sampling at Every 50^0

The image presented to the IPCA network as described in the methodology. The recognition is achieved by finding th e minimum distance between the coefficients of a test image and the training images. The vector of the image in the training set that is nearest to the test image non Gaussian vector is chosen as the recognized image.

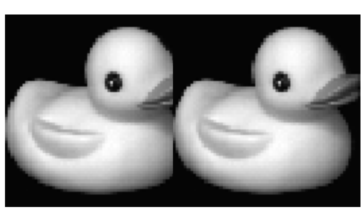
Training images are sampled by 50^0 in pose angle. That is to say, there are 6 training images for each object, making a total of 140 training images. The images that were not in the training set were considered as test images, thus making a total of 1300 test images. In the following two experiments, number of non Gaussian vectors q is turned parameters q= {20, 25,30,35,40, 45, 50} and the Table II shows the results and the Fig 5 represents the test applied to that object



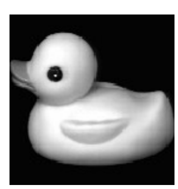
(a)



(b



(c)



(d)

Fig 5: Object Extracting Using IPCA-ICA (50^0)
(a) Input image (b) Object is mixed with some unknown mixtures
(c) Object Extracting Using IPCA-ICA (d) Duck with Orient 150^0

Number of eigenvectors used							
	20	**25**	**30**	**35**	**40**	**45**	**50**
ICA	1000	990	980	970	960	970	960
PCA	990	980	960	960	950	940	930
IPCA-ICA	1000	1010	1020	1030	1030	1000	1000

Table I Number of correct recognitions by using ICA, PCA and IPCA. The Pose angle sampling is 50^0 . The recognitions are shown for a Total of 1280 test images

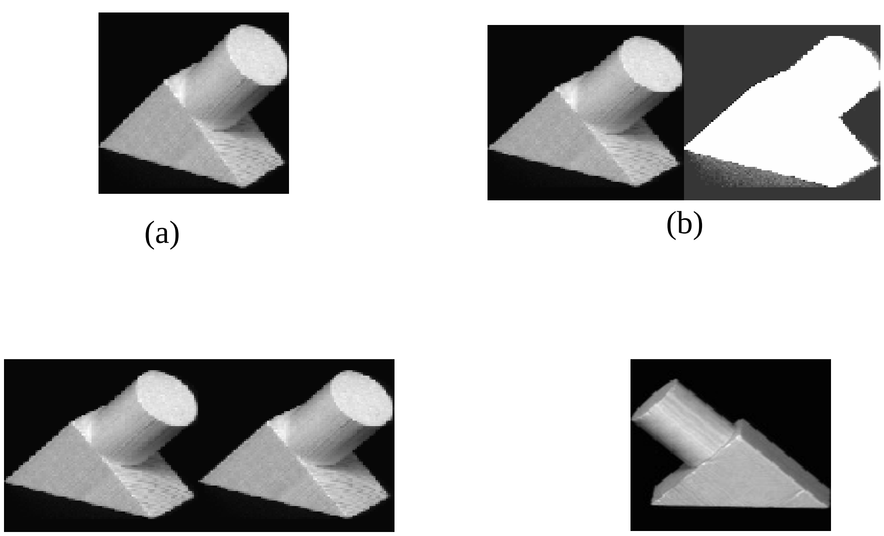
3.2. Results with Pose Angle Sampling at Every 25^0

The previous experiment is also repeated by using images sampled at every 25^0. This gave a total of 300 training objects, and the rest, 1140 images as the test objects. In this case, there is more information for the network to learn. The performance in both the PCA, ICA as well as the IPCA is increased. Number of non Gaussian vectors as follows: q= {20, 25,30,35,40, 45, 50}

Number of eigenvectors used							
	20	**25**	**30**	**35**	**40**	**45**	**50**
ICA	970	1090	900	910	944	950	930
PCA	980	980	910	920	900	980	920
IPCA-ICA	1090	1000	1020	933	990	1000	1000

Table II Number of correct recognitions by using ICA, PCA and IPCA. The Pose angle sampling is 25^0. The recognitions are shown for a Total of 1140 test images

From the above figures, we can see the, present method produced better results, and the recognition rate has got a significant increasing compared with PCA. We can come to the conclusion that the performance of IPCA outperformed the linear PCA. But the Incremental PCA is recursive method and the non Gaussian vectors is calculated for each image and the non dominant vectors are not considered for the next stage so this method is best suitable for 3D object recognition, performance of 87.88 recognition rate is obtained when Number of non Gaussian vectors equals to 7, the average success rate for the PCA, ICA and IPCA-ICA methods is shown in Table III and the performance of PCA , ICA ,IPCA-ICA methods in the 5 nearest match is shown in Fig 6.



(a) (b) (c) (d)

Fig 6: Object Extracting Using IPCA-ICA (25^0)
(a) Input image (b) Object is mixed with some unknown mixtures
(c) Object Extracting Using IPCA-ICA (d) Second with Orient 25^0

	COIL 1(25^0)	**COIL2(50^0)**
PCA	82.58	74.88
ICA	83.88	76.22
PCA+ICA	83.23	75.55
IPCA_ICA	87.88	79.12

Table III Average success rate for COIL objects database

3.3. Results with Pose Estimation

In this experiment, we are interested in modeling the nonlinear manifold formed by the object appearance under varying poses. The manifold is embedding into the Low-dimensional subspace as follows. If the pose angle parameter is a continuous variable, the manifold will be a smooth curve. Since it is not possible to capture images with the pose angle as a continuous variable, the manifold will appear as a piecewise continuous curve, and if the pose angle is fine enough, the curve will appear to be smooth. In our experiment, we use 15 images for learning, one at every 25 degrees of rotation. The remaining 57 images are left for evaluating the pose.The manifold for the first object is shown in Fig 7. The non Gaussian space is constructed by using sampled images. The curve appears smooth with the markers showing the location of the non Gaussian image in the 3-D non Gaussian space with ql, q2, and q3 being the three most dominant non Gaussian vectors..

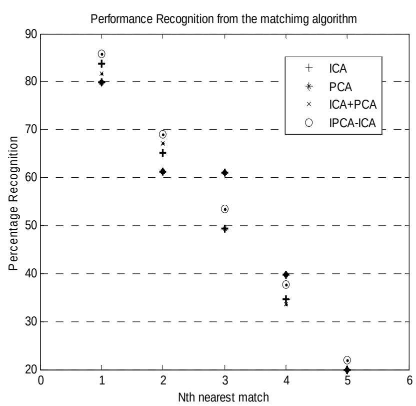


Fig 7 Percentage recognition with nth

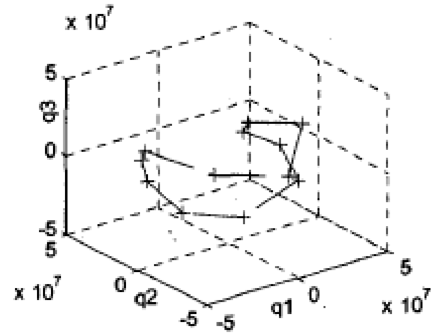


Fig 8 Appearance manifold of the first object

Nearest match (The pose angle is sampled at every 25^0 .)

Also, the recognized curve seems close to the object, since the last view of the object is almost the same as the first one. Testing points for each of the object manifolds are obtained through the projection of the training images onto the non Gaussian space .The manifolds are then finely sampled through cubic spline interpolation.

Finding the nearest object manifold as described in the previous paragraph provides an estimate of the object pose. In addition to the universal non Gaussian space that describes the variation in all the images of all the objects of the database, a separate non Gaussian space for each object is constructed too. The universal non Gaussian space is used for object identification and the individual object non Gaussian space is used for pose estimation. The manifolds obtained in the object non Gaussian space can be parameterized by object pose.

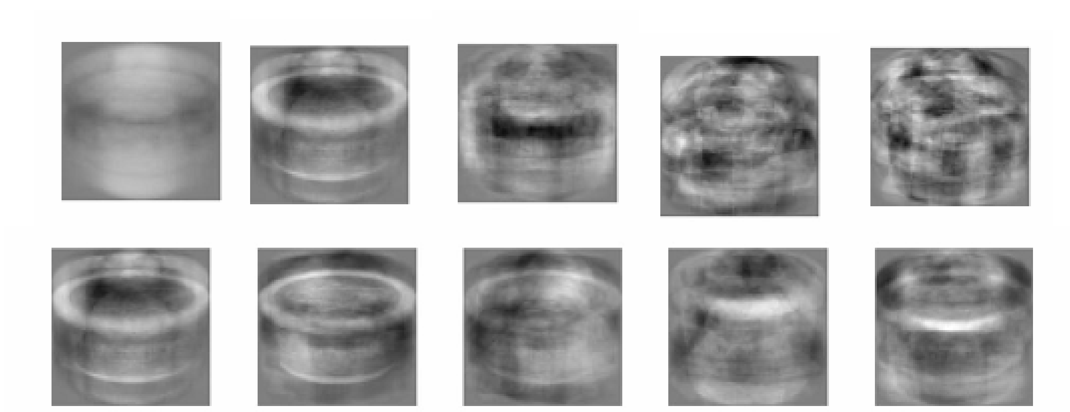


Fig 9 the first 10 efficient non-Gaussian objects taken from the COIL Database

4. CONCLUSIONS

In this paper, a new feature extraction method for object recognition tasks based on incremental update of the non-Gaussian independent vectors has been used. This method concentrates on a challenging issue of computing dominating non-Gaussian vectors from an incrementally arriving high-dimensional data stream without computing the corresponding covariance matrix and without knowing the data in advance, and the results are to be compared with PCA and ICA. Three experiments were performed with different pose and non Gaussian vectors.

The images of the COIL database have been originally used by many people for testing the appearance-based recognition system, based on the notion of parametric non Gaussian space Our results seem to compare favorably with respect to the results reported in [1][2], Note that IPCA not only allows for the construction of training images of much smaller size, but also can identify the object pose. Experiment results in appearance-based 3D Object Recognition confirm IPCA offer better recognition rates.

REFERENCES

"Visual learning and recognition of 3-d objects from appearance", Int. J. Computer Vision, vol. 14, no. 1, pp. 5–24, 1995.

A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis: Wiley, 2001.

C. Jutten and J. Herault, "Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture," Signal Processing, vol. 24, no. 1, pp. 1–10, 1991.

cran.r-project.org/doc/packages/**fastICA**.**pdf**

Dr. V.K Ananthashayana,"Face Detection using ICA with DCT faces" International conference on IT Nepal-2003

Harkirat S. Sahambi and K. Khorasani, "A Neural-Network Appearance-Based 3-D Object Recognition Using Independent component Analysis", neural networks, vol. 14, no. 1, 2003

http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Issam Dagher and Rabih Nachar. "Face Recognition Using IPCA-ICA Algorithm", IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 6, 2006.

J. Karhunen and J. Joutsensalo, "Representation and Separation of Signals Using Non Linear PCA Type Learning," Neural Networks, vol. 7, no. 1, 1994.

J. Rubner and K. Schulten, "Development of Feature Detectors by Self-Organization," Biological Cybernetics, vol. 62, pp. 193-199, 1990.

K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed. NewYork: Academic Press, pp. 831-836, Aug. 1996.

M. A. Turk and A. P. Pentland, "Face recognition using Eigenspaces," in *Proc. IEEE Computer Society Conf. Computer Vision Pattern Recognition*, Maui, HI, 1991, pp. 586–591.

M.Hafed and martin D. Levine, "Face Recognition Using the Discrete Cosine transform International Journal of Computer Vision 43(3), 167–188, 2001

Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski, "Face Recognition by Independent Component Analysis", Neural networks, vol. 13, no. 6, November 2002

P. Common, "Independent Component Analysis, a New Concept?" Signal Processing, vol. 36, no. 3, 1994.

S. A. Nene, S. K. Nayar, and H. Murase. (1996, Feb.) Columbia object image library (coil-20). Columbia University. [Online] http://www.cs.columbia.edu/CAVE/

T. Poggio and S. Edelman, "A network that learns to recognize 3D objects," Nature, vol. 343, pp. 263–266, 1990.