Eigenimage analysis for object recognition

Ovidiu Ghita, and Paul Whelan

Vision Systems Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland
{ghitao, whelanp}@eeng.dcu.ie
http://www.eeng.dcu.ie/~whelanp/vsg/vsgper.html

Abstract. A method for object recognition and pose estimation is presented. The approach discussed is a variant on current approaches to eigenimage analysis. Compared to traditional approaches which use object geometry only (shape invariants), the implementation described uses the eigenspace determined by processing the eigenvalues and eigenvectors of the image set. The image set is obtained by varying pose whilst maintaining a constant level of illumination in space, and the eigenspace is computed for each object of interest. For an unknown input image, the recognition algorithm projects this image to each eigenspace and the object is recognised using space partitioning methods which determine the object and the position in space. Several experimental results have been obtained to demonstrate the robustness of this method when applied to the robotic task.

Keywords: 3-D, eigenspace, object recognition, image set, object set

1 Introduction

One of the main aims of an intelligent robotic vision system is to recognise objects and to compute their position in space. The vision recognition task must be performed as close to real time as possible. Therefore the recognition algorithm must be computationally inexpensive.

In this paper we present an approach based on the use of eigenvectors (eigenimage). This can be briefly described as a method, which computes the eigenspace determined by processing the eigenvalues and eigenvectors of the image set (see also [6], [11], [12], [18], [22]). For our practical implementation in order to decrease the number of images, the image set is obtained by varying pose while maintaining a constant level of illumination. The eigenspace is determined using the eigenvalues (eigenvectors) of covariance matrix (i.e. a symmetric matrix) in order to obtain a low dimensional subspace.

For an unknown input image, the recognition algorithm projects this image to eigenspace and the object is recognised using a space partitioning method, which

determines the object and also its position in space according with the number of images that describe the position from the image set [10].

The image set is normalised in brightness and the background noise removed in order to eliminate the redundant information. The eigenspace for the image set is built by computing the largest eigenvalues (eigenvectors) of the set. The next step is to project all the images onto eigenspace and we obtain a set of points that characterise the object and the pose in space. In order to increase the resolution for position estimation we connected these points in eigenspace and an unknown position can be better approximated. The first step could be considered by analogy with neural networks as "learning" and the recognition by space partitioning as "matching".

In our approach we assume that the objects are not occluded. Another important problem can be the texture reflectance and this problem is directly connected to the illumination conditions.

In comparison with other approaches based on geometry recognition (see also [1], [8], [19]) this method is suitable for a large number of objects with different shape geometry.

2 Eigenspace technique

The image I(x,y) is represented by a two-dimensional 256 by 256 array of 8-bit intensity values. This image can also be considered as a vector of dimension 65536. This is obtained by reading pixel brightness values in a raster scan manner.

To reconstruct an image we map a collection of points (in our case eigenvalues) in this huge space. The main idea of the principal component analysis (Karhunen-Loeve expansion) is to find the vectors that can better describe the image (or the entire space).

These vectors describe an orthonormal space and this property is presented below:

$$u_i u_j^T = g_{ij} = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases} \tag{1}$$

where u_i, u_j are two eigenvectors and g_{ij} is the scalar product.

The image I can be represented as:

$$I = [i_1, i_2, i_3, \ldots, i_N] \tag{2}$$

where $i_1, i_2, \ldots, i_N$ are the pixel values.

This vector represents an unprocessed image. The reflectance proprieties of the object shape can vary from scene to scene. If the illumination conditions of the environment are maintained constant, the appearance is affected only by object position.

The image set for an object is obtained as:

$$[I_1, I_2, I_3, \ldots, I_P]^T \tag{3}$$

where P is the number of considered positions in space.

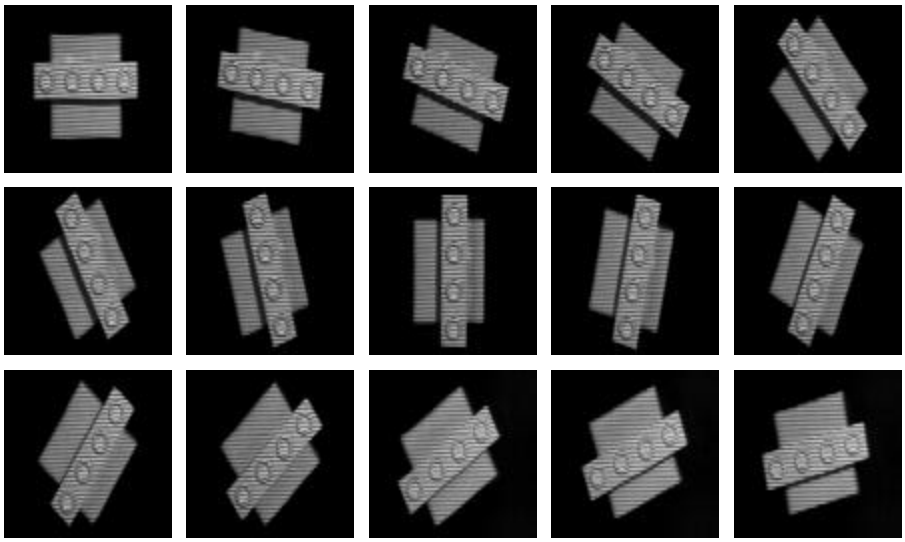


Figure 1. Image set obtained by rotating a sample object. All images are background normalised.

Because we do not want our images to be affected by the variations in the intensity illumination or the aperture of the imaging system (lens), we normalise each image so that the total energy contained in the image is unity [10]. The normalised image is obtained by dividing each pixel by the following number:

$$i_n' = \frac{1}{B} i_n, \qquad B = \sqrt{\sum_{n=1}^{N} i_n^2} \tag{4}$$

where n represents the pixel's index and N is the dimension of the image.

Before computing the eigenspace we need to calculate the average image (A) of all images:

$$A = \frac{1}{P} \sum_{i=1}^{P} I_i' \tag{5}$$

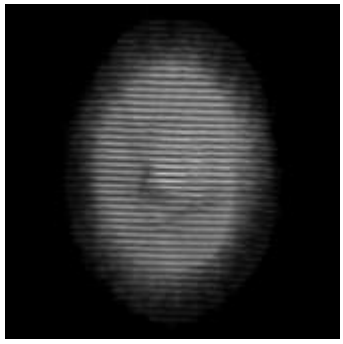


Figure 2. The average matrix for image set.

The image set will be obtained by subtracting the average image (A) from each normalised image:

$$S = [I_1' - A, I_2' - A, I_3' - A, \ldots, I_P' - A]^T \tag{6}$$

The dimension of matrix *S* is P x N, where P is the number of positions (images) and N the number of pixels. The next step involves the computing the covariance matrix:

$$C = S^T S \tag{7}$$

This matrix is very large (65536 x 65536) and it will be extremely difficult to compute the eigenvalues and the corresponding eigenvectors. If the number of images P is smaller than N it is easier to construct the P by P matrix $Q=SS^T$, but in this case the dimension of space is maximum P. The eigenvalues and the corresponding eigenvectors are computed solving the following well known equation:

$$Qu_i = v_i u_i \tag{8}$$

where u_i is the i^{th} eigenvector and v_i is the corresponding eigenvalue. In this implementation the eigenvalues and eigenvectors of the covariance matrix are

computed using the Householder algorithm in order to obtain a simple tridiagonal form. This is followed by an application of the QL algorithm [16].

The eigenspace is obtained by multiplying matrix of eigenvectors (eigenmatrix) with the matrix S:

$$E = US \tag{9}$$

where $U=[u_1,u_2,\ldots,u_P]^T$, U is P x P dimensional. As we expect the matrix E that represents the eigenspace is P x N dimensional.

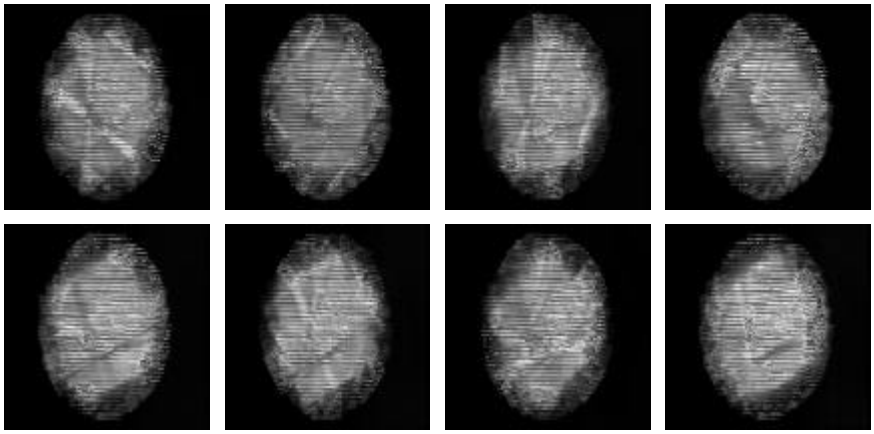


Figure 3. Eight of the eigenvectors corresponding to the largest eigenvalues calculated for the input image set.

Using this approach the number of calculations is significantly reduced but in this case the number of eigenvectors is relatively small (up to 36). For an exact description we need N eigenvectors, but for recognition purpose a smaller number is sufficient to describe a low dimensional subspace.

3 Object recognition and position estimation

Once the eigenspace is computed we can project all images from the set on this subspace ($E=[e_1, e_2,\ldots,e_p]^T$). The result will be a collection of points that describe the object and its position. Before projecting the image set onto eigenspace, we subtract the average image from the image set.

$$h_i = [e_1, e_2, \ldots, e_P]^T (I_i^{'} - A) \tag{10}$$

where $e_1, e_2, \ldots, e_p$ are the eigenspace vectors and each vector is N dimensional.

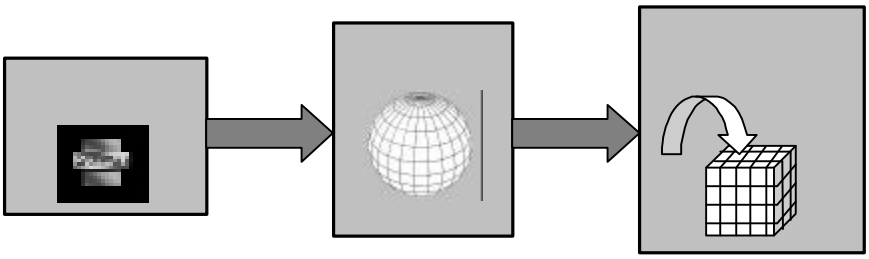


Figure 4. Building the eigen database

As we expect each point is P dimensional and this appears to be a sever restriction in image reconstruction. But for the purpose of recognition, this allows us a fairly accurate method under the condition of maintaining a constant illumination. If these points are connected in P space to give us the possibility of estimating positions of the object that are not included in the image set. A direct connection between position of the object and the projection on the eigenspace (or object space) is obtained.

An unknown input image will be projected onto eigenspace and as result we will obtain a P dimensional point. This vector can be used in a standard recognition algorithm and the simplest method to determine the best matching is to compute the Euclidean distance [10], [21].

$$d_i^T = \left\| h - h_i \right\|^2 = (h - h_i)^T (h - h_i) = h^T h - h_i^T h_i \qquad (11)$$

where $i=1,2,\ldots,P$ and h is the projection of the input image onto eigenspace. A better representation in space is offered by Mahalanobis distance (for more details see [20]), which is related to the image set, but for the sake of computational efficiency the Euclidean distance was used. The Mahalanobis distance is given by:

$$d^2 = (h - h_{m,X})^T C_X^{-1} (h - h_{m,X}) \qquad (12)$$

where $h_{m,X}$ is the mean of the class X and C_X^{-1} is the inverse of the covariance matrix. The use of the Mahalanobis distance removes several limitations of the Euclidean distance, such as a better correlation between features (in our case P-dimensional points) that belong to different classes and is invariant to any nonsingular linear transformation. If the features are uncorrelated in this case the Mahalanobis distance becomes similar to the Euclidean distance.

We consider the recognition task to find the object and its position. The object is in the collection when the object gives us the minimum distance and this distance is smaller than a threshold value.

$$d_i = \min \left\| h - h_i \right\| \le J \qquad (13)$$

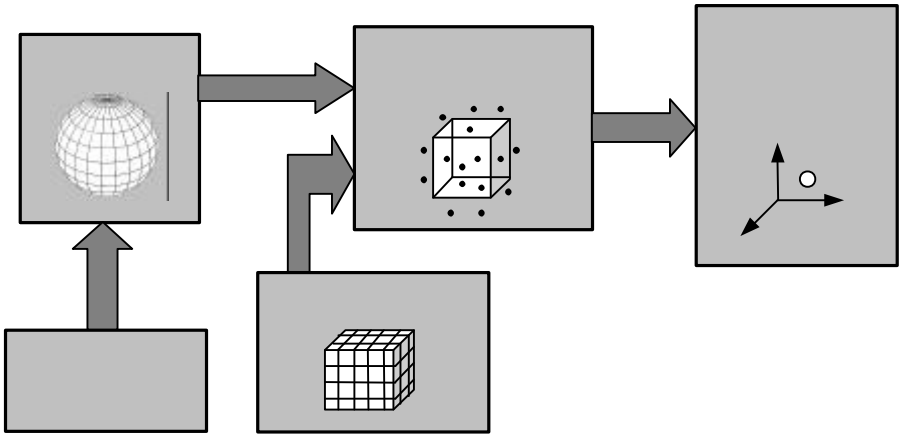


Figure 5. Data-flow diagram of the object recognition system.

If we have a large collection of objects the best approach is to compute the universal eigenspace which contains all images and all positions. The first step is to recognise the object projecting the input image on this universal eigenspace and after this we can estimate precisely the position by projecting the input image on the object eigenspace.

3.1 Space partitioning technique

For a large database a better solution is an algorithm based on space partitioning. *Hashing and indexing* techniques are the simplest implementation, but the index table increases exponentially with the dimension of space. Space partitioning techniques provide a practical solution for multi-dimensional search implementation. One of the most popular space partitioning technique is the *k-d tree* developed by Bentley and Friedman [4]. The *k-d tree* structure partitions the space using the hyperplanes perpendicular to the co-ordinate axes. The database has a root node, and the rest of the points are lying on one hyperplane. All the points in database are connected by passing through the root node, the new points are added to the left child (the left child is the point derived from the root node). This process is applied on the left to right until all the points from database are classified. This algorithm is illustrated in the figure 6.

In order to simplify the representation in figure 6, the space is partitioned only for the first three co-ordinates *(x,y,z)* while the recognition algorithm uses a multi-dimensional space (up to 36). The candidate list structure is organised into a database, the method uses a 1-D binary searches to find the points between a pair of parallel hyperplanes.

In figure 6, a correct matched point (for a class of objects) is inside the cube of size 2ϑ. The first step of the algorithm is to recognise the object projecting the input image on this universal eigenspace. After this, we can estimate precisely the position by projecting the input image on the object eigenspace (the position in space is determined by the point position in the hypercube)

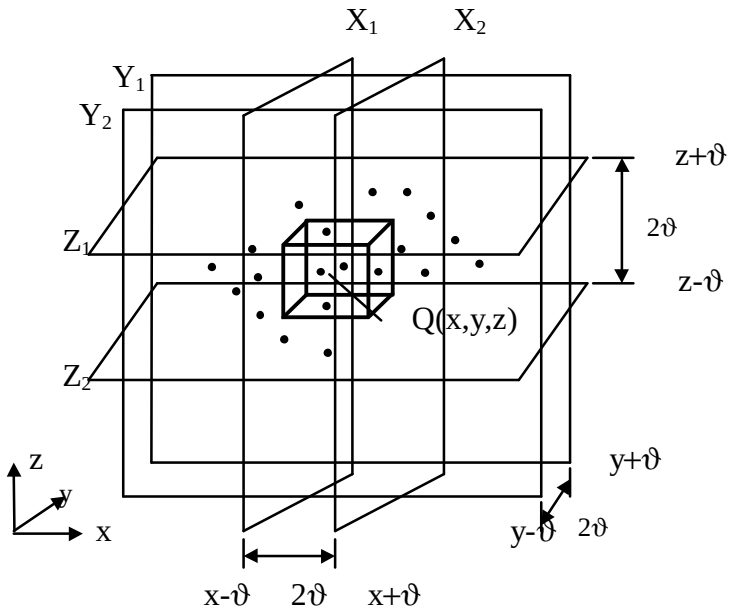


Figure 6. The space partitioning algorithm (based on [15]).

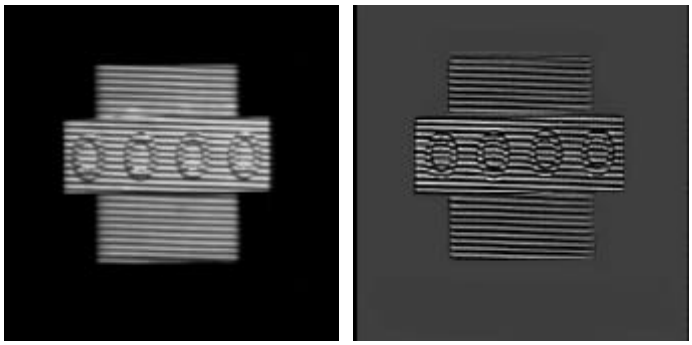
Certainly the universal eigenspace will be difficult to compute since the number of images for a large collection of objects is huge. For this reason we will compute the largest P eigenvalues and the corresponding eigenvectors. But the advantage in recognition terms is very important because we can determine the object from collection in only one step (see also [2], [9]). The second step is to estimate the position of the object.

Otherwise we have to search each object's eigenspace one by one and this method is far from efficient.

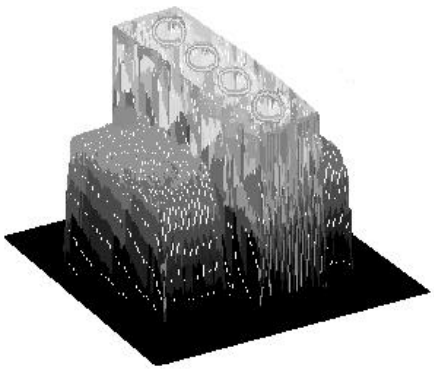
4. 3-D object recognition

We intend to use this approach for our bin picking application. In order to recognise the object and its position in space is necessary to have a depth perception of the scene, allowing the comprehension of 3-D relationship between objects in the real world. For this purpose we use a variant of the depth from defocusing technique

originally developed by Pentland. The depth estimation is calculated from the degree of blurring in the two images (more details can be found in [14], [17]). This transition from 2-D image to 3-D image is presented in figure 7.



(a) (b)



(c)

Figure 7. (a) 2-D image, (b) 3-D depth map, and (c) the spatial representation of the 3-D depth map

Certainly in this case it is necessary to construct the universal eigenspace and each objects eigenspace using images corresponding to the depth map. The recognition task remains unchanged by projecting the input image on the eigenspace and compute the Euclidean distance.

5 Experiments and results

Designing a practical system for object recognition require accuracy and speed. In industry 75% of applications are in small to medium batches [3]. Also, Delchambre [5] highlights the fact that 98% of products are made of fewer than 25 parts.

Therefore, if the number of objects and corresponding positions are small (less than 100), the best implementation is to construct only a universal eigenspace. Clearly this is an approximate position estimation for an object set that contains more than 4 objects, but suitable for a range of applications. If the accuracy or the number of objects is large we are required to construct the universal eigenspace and the object eigenspace for each object [10]. In our implementation we used both methods.

Another key problem is the dimension of the eigenspace. As we discussed before this number is limited to P which is the number of positions for each object. This number can vary from object to object and is in direct connection with the number of objects. For this present algorithm we have obtained poor results if the dimension of the eigenspace is less than 8. We increase the eigenspace dimension to 36 step by step but the rate of recognition is not affected visibly after 15 when the recognition rate is near to 100%.

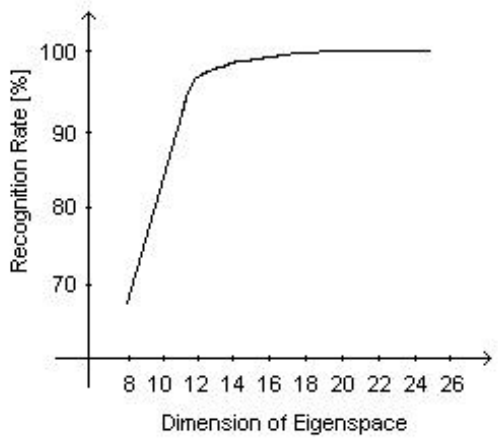


Figure 8. Recognition rate as a function of the dimension of eigenspace.

A key issue is maintaining the illumination at a constant level. This requirement is very important especially for 3-D recognition, as the depth map is sensitive to different levels of illumination. For 3-D detection we project a structured light pattern to obtain a better representation of shape. Object reflectance and occlusion can cause errors in the recognition process (more details concerning object reflectance can be found in [13]). If the level of light is not constant for each position, then we have to acquire the same image with different degree of illumination. It is clear that

in this case the number of images is huge and to solve this problem is computationally intensive.

In our implementation we used a set of 5 objects that are presented in figure 9. All objects contained in the object set have almost the same colour and reflectance properties but different shapes.

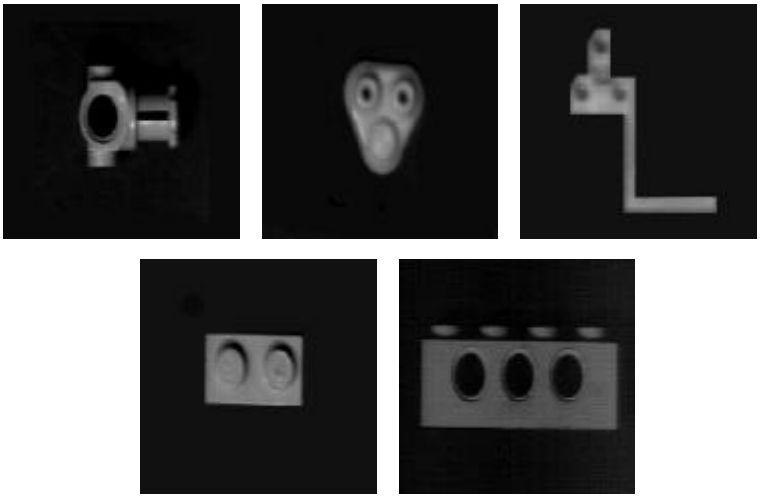


Figure 9. Object set. All images are background normalised

As a result we have obtained correct recognition for objects from an image set (universal image set) and position estimation with an error of less than 5 degrees for an object set which contains 36 images. If the object from image set has a small number of positions (less than 10), the results concerning pose estimation are unreliable.

6. Conclusions

We have presented a possible approach to object recognition for robotic vision applications. In comparison with other methods this implementation is simple, fast and reliable even if it is not a unique solution for recognition problem (also see [7], [8], [19]).

It is very important to remember that many applications do not require unity rate of recognition for position and only estimation with a certain error. Furthermore this approach can be easily implemented in hardware for a real time application and the future research will be focused in this direction (for more details see [15], [21], [22]).

Our experiments show that the eigenspace technique can be used for object recognition and position estimation with a very high degree of accuracy. Also a novel

approach for 3-D object recognition was briefly presented. All these results prove the robustness and recommend this method for robotic applications.

Acknowledgements

This research was supported by Motorola BV, Dublin, Ireland.

References

1. P. Besl and R. Jain, "Three-dimensional object recognition", *ACM Computing Surveys*, 17(1), pp. 75-145, 1985
2. T. Bailey and A. K. Jain, "A note on distance-weighted k-nearest neighbour rules", *IEEE Trans. Syst. Man and Cybern.*, vol. 8, no. 4, pp. 311-313, 1978
3. B.G. Batchelor, "Intelligent Image Processing in Prolog", *Springer-Verlag*, London, 1991
4. J.L. Bentley and J.H. Friedman, "Data Structures for range searching", *Computing Surveys*, vol.11, no. 4, pp. 397-409, Dec. 1979
5. A. Delchambre, "Computer-Aided Assembly Planning", *Chapman & Hall*, 1992
6. O. Ghita and P.F. Whelan, "Object recognition using eigenvectors", Proceedings of SPIE, vol. 3208, pp. 85-91, Pittsburgh, Oct. 1997
7. K. Ikeuchi, "Generating an interpretation tree from a CAD model for 3-D object recognition in bin-picking tasks", *Inter. J. Comp. Vision*, vol.1, no.2, pp. 145-165, 1987
8. D. Kriegman and J. Ponce, "On recognizing and positioning curved 3-D objects from image contours", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI 12, no. 12, pp 1127-1137,1990
9. J. Macleod, A. Luc and D. Titterington, "A re-examination of the distance-weighted k-nearest neighbour classification rule", *IEEE Trans. Syst. Man and Cybern.*, vol. 17, no. 4, pp. 689-696, 1987
10. H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance", *International Journal of Computer Vision*,14, pp. 5-24, 1995
11. B. Moghaddam and A. Pentland, "Face recognition using view-based and modular eigenspaces", *Automatic Systems for the Identification and Inspection of Humans*", SPIE, vol.2277,1994
12. H. Murakami and V. Kumar, "Efficient calculation of primary images from a set of images", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI 4, no. 5, pp. 511-515, 1982
13. S. K. Nayar and R. Bolle, "Reflectance based object recognition", *Inter. J. Comp. Vision* vol. 17, no.3, pp.219-240,1996
14. S. K. Nayar, M. Watanabe and M. Noguchi, "Real-time focus range sensor", *Columbia University, Tech. Rep. CUCS-028-94*, 1994

15. S. A. Nene, S. K. Nayar, H. Murase, "SLAM: A software Library for Appearance Matching", *Proc. of ARPA Image Understanding Workshop*, Monterey, Nov. 1994
16. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, "Numerical recipes in C", *Cambridge University Press*, 1992
17. A. P. Pentland, "A new sense for depth of field", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-9, no.4, pp. 523-531, 1987
18. L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces", *J. Opt. Soc. Amer.*, vol. 4, no.3, pp. 519-524, 1987
19. F. Stein and G. Medioni, "Structural indexing: efficient 3-D object recognition", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI 14, no. 2, pp. 125-145,1992
20. C.W. Therrien, "Decision estimation and classification. An introduction to Pattern Recognition and Related Topics", *John Wiley & Sons*, 1989
21. M. Turk and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, pp. 71-86, 1991
22. M. Turk and A. Pentland, "Face recognition using eigenfaces", *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.586-591, June 1991