Real-Time Gesture Recognition using Eigenvectors

Vaughn M. Segers, James Connan
Department of Computer Science
University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa
Telephone: +(27) 21 959-3010, Fax: +(27) 21 959-3006
Email: 2435465@uwc.ac.za, jconnan@uwc.ac.za

Abstract—This paper discusses an implementation for gesture recognition using eigenvectors under controlled conditions. This application of eigenvector recognition is trained on a set of defined hand images. Training images are processed using eigen techniques from the OpenCV image processing library. Test images are then compared in real-time. These techniques are outlined below.

Index Terms—hand shape, gesture recognition, eigenvectors, sign language

Fig. 1. Various objects which have been recognized using eigenvector methods such as faces and rotated letters.

I. INTRODUCTION

THIS investigation considers the area of whole-hand gesture recognition in real time. Areas such as interactive displays, sign language recognition and home automation are some of the applications of such technology.

The objective was to create a system which could detect various hand shapes and hand orientations and identify them for the user in real time. A fast matching system is thus required. For this reason the eigenvector method was chosen.

Eigenvectors have been widely used in facial recognition systems, seen in [4][5][6][7]. The area of facial recognition using eigenvectors has been widely researched since the 1980s[4][5] up to recent times[2].

There has also been research in the application of eigenvectors to the area of sign language recognition[8]. As such this work extends on this area by the real-time application of this technology.

This paper will cover the following areas:

1) We review the current approaches to gesture recognition, both for the hand and for the entire body.
2) We discuss the use of eigenvectors and their various areas of use in recognition.
3) The pre-development issues are determined.
4) We then outline the system development considerations.
5) We then describe the experimental setup of the system, from training to testing phase.
6) Test results are given.
7) The conclusion and ideas for future work are presented.

II. LITERATURE REVIEW

A. Gesture Recognition

Implementations of gesture recognition systems will be reviewed below.

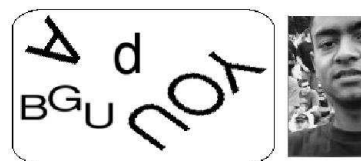Black and Jepson[8] developed a novel system called Eigentracking. Within their research they studied how eigenvectors could be used to identify and track articulated objects within a scene. Their work focused on the tracking of arbitrary objects, in this case, a soda can. Their work was also extended to show the effectiveness of the system on a set of four hand signs. Their results show an excellent level of accuracy on the trained gestures.

Pentland et. al. [9] produced a real-time sign language recognition system which relied on Hidden Markov Models. Continuous signs in the whole body view were investigated. Their work revealed a 92% to 98% accuracy on their 40 word lexicon. This illustrates that Hidden Markov models can be used as an effective system for real-time gesture recognition.

Cooper and Bowden [2] describes the recognition of sign language using Markov Chains. The difference with this application of the Markov Chains is the pre-processing performed on the image. Face detection, combined with a grid based sign classifier,is used to segment the image into identifiable regions. Their system achieves a 74.3% level of accuracy on their 164 word lexicon.

B. Eigenvector Approaches to Matching

Eigenvector approaches have been used in many areas of recognition, such as facial, text, and object recognition, as in figure 1.

Leonardis and Bischof[3] use the eigenvector approach[1] to find objects within a scene. Specifically they are interested in finding rotated objects within a noisy scene. Their work also covers the rotations and occlusions experienced by the object under investigation. Though these problems of occlusion and rotation are solved, no mention is made of attempting to create a real-time system. The approach used however is novel and provides accurate recognition.

---

[1]They term this the 'eigenspace' approach. These terms are used interchangeably. Those who use faces, term it 'eigenfaces'. Those who use letters, term it 'eigenletters', etc.
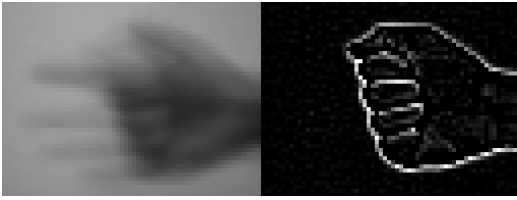
Fig. 2. An example of the average image calculated on the left and the negative image on the right, otherwise known as the mean normalized column vector

The earliest use of eigenvectors for image recognition is in the field of face recognition[4]. Many systems have been built for this purpose[6][7] and have proved successful.

Sirovich and Kirby produced a system in 1986[5] which recognized faces using the eigenvector approach. They show that any face can be recognized by a low-dimensional feature space when using the eigenvetcor approach. It is this characteristic of the eigenvector approach that lends itself to real-time recognition systems.

Pentland, Moghaddam, and Starner [6] took this concept further in demonstrating that facial recognition held true even in varying orientations of the face, as well as changes in facial expression. Their system achieved recognition rates between 95 and 98 percent on an image database of 3000 images.

To further emphasize the application of the eigenvector method, we look at the work of Hase et. al. [1]. Their work recognized rotated alphanumeric characters from pictures. Good recognition was achieved in the 26 capital letters of the English alphabet in the Century font style. The methodology for all eigenvector methods are similar and will be discussed in the sections to follow.

III. EIGENVECTOR THEORY

The theory behind all eigenvector systems remains the same. The purpose of which is to reduce the size of the images to be recognized from a high to a lower dimension. While lowering the dimensionality in the set, using the eigenvector method also highlights the variance within the set. This is achieved in the following manner:

By performing singular value decomposition on an average covariance matrix we find the eigenvalues and eigenvectors as in equation 1.

$$Cu = \lambda u \qquad (1)$$

Where C is the average covariance image matrix, u corresponds to the eigenvectors and λ corresponds to the eigenvalues of C. The creation of the average covariance matrix C will be discussed later. We now need to find the average image ψ of all the images in our image set. This is determined by summing all the images in our image set and dividing by the number of images in the set. An example of the average image can be seen in Figure 2. The equation for finding the average image is equation 2.

$$\psi = \frac{1}{M} \sum_{k=1}^{M} \Gamma_i \qquad (2)$$

The mean normalized column vector ϕ_i is now determined for every image k in the image set. This is done by first reformatting the image in the standard lexicographical order to become a column vector Γ_i. M is the number of images in the set.

The previously calculated average image ψ is then subtracted from the column vector Γ_i to produce the mean normalized column vector ϕ_i. ϕ_i is a mathematical representation of the differences between the set and the image.

$$\phi_k = \Gamma_k - \psi \qquad (3)$$

Now to find the necessary average covariance matrix C we:

$$C = \frac{1}{M} \sum_{k=1}^{M} \phi_k \phi_k^T \qquad (4)$$

When displayed, the mean looks like a negative of the image. This negative represents all of the variance within the set. An example of this negative image can be seen in Figure 2. Now that C is determined we can compute the eigenvalues and eigenvectors for the image set. The eigenvectors which correspond to the highest eigenvalues represent the most expressive features of the image set. The chosen eigenvectors can estimate any image in set. We now move to compute the principal components. The dot product of every image and the chosen eigenvectors are found. This dot product is known as the principal component. This value will uniquely identify each image.

Recognition is performed by finding the dot product of the new image and the stored eigenvectors. The euclidean distance was chosen as the comparison measure between images. The shortest euclidean distance between the new and saved principal components indicates the closest match to the new image.

A matched hand gesture must also fall within the matching threshold. If the shortest euclidean distance found between an image and the training set is outside the threshold, the hand is either not in the training set or the image is not a hand.

IV. DEVELOPMENT

The system required extensive use of mathematical libraries as evidenced in Section III.

The OpenCV Computer Vision Library was found to have eigenvector classes geared towards use in image processing. Specifically these libraries were used for facial recognition.

OpenCV was used in the development of the final system. The OpenCV Computer Vision Library has been created to accept any image input format. It is for this reason that training images were manually taken from the native image capture application in Ubuntu Linux and used in the system.

The system is divided into two sections, the training phase and the testing phase.

A. Training Phase

The training phase follows the following steps:

1) During the training phase, the images in the training set are loaded onto the system. These images comprise of those found in figure 3.

2) First, as described in section III, we compute the average image for all the images in the training set. An example of this average image can be seen in figure 2. The system now has every image loaded as well as the average of those images.

3) Using this information the OpenCV function is called to calculate the eigenvectors and eigenvalues. This is all the information required to determine the principal components of the training set.

4) The principal components can now be calculated and used to judge the closest match to the training images.

5) OpenCV then outputs the generated principal components, eigenvectors and eigenvalues as training data to an XML file for further use in the Testing Phase. The principal components uniquely partition each hand shape into a different subspace.

B. Testing Phase

1) The XML file produced in the Training phase is called during the testing phase. This data is used to identify new images.

2) The eigenvectors are loaded from the stored data.

3) The dot product of each new image and the stored eigenvector is calculated. This produces a principal component unique to that image.

4) The euclidean distance is checked between the newly calculated principal component and the stored principal components. This is done to determine the similarity to the trained images. The shortest distance corresponds to the letter class to which the gesture belongs.

The drawback to the eigenspace method is the time for loading of images and the XML file. The more images are used, and the greater their size, the longer it will take for the training phase to complete. The testing phase will also take longer to initialize as the XML file also grows with the size of the image set. Larger image sets will require greater memory space to run the program. This however has minimal impact on the time it takes to recognize gestures or the real-time implementation of the system.

V. EXPERIMENTAL SETUP

The system is run on a Sony Vaio Intel Centrino Duo 1.2GHz notebook computer with 2GB RAM, running Ubuntu Linux 8.04. The camera used is a Logitech Quickcam Chat. The images were captured at a resolution of 640x480 pixels. Video was captured at the same resolution.

Signs were trained and tested in front of a noise-free background as in figure 4. The test bed consists of the nine sign language classifiers.
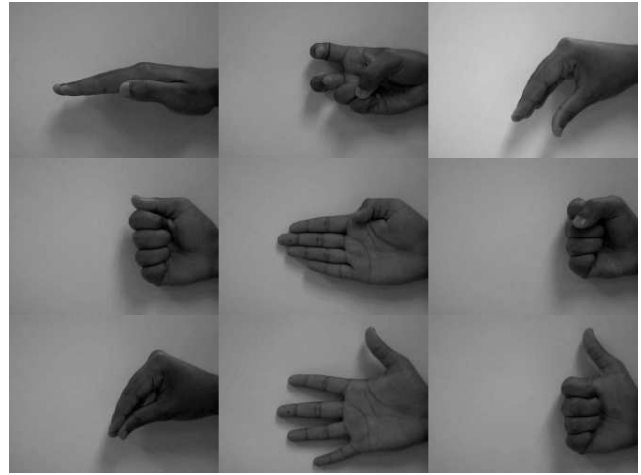

Fig. 3. The nine different South African Sign Language classifiers identified by the Thibologa Institute. From left to right. 1) Top Row: Palm/Flat, Two Long Thin Bent Extensions, Narrow/Shallow flat object 2) Middle Row: Round/Sperical Object, Long Flat Smooth Surface, Fist 3) Bottom Row: Flat/Triangular Object, Index Five, Compact Mass with Salient Extension
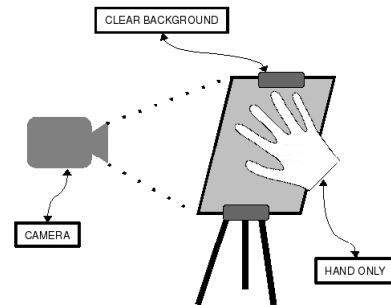

Fig. 4. The experimental setup for training and testing the system

The camera was placed at a stationary position in front of the signer. No attempt was made to ensure that the lighting conditions remained constant at the time of training or testing.

The signer was required to only place their hand within the view of the camera and no other body part. As far as possible the wrist of the signer was not included in the training data.

Signs for the testing phase were taken from the Thibologa Sign Language Institution's information booklet. The signs used are sign language classifiers as seen in Figure 3.

The signer for the testing and training data had no previous experience in South African Sign Language.

VI. TEST RESULTS

Testing was done by having the user sign the previously trained nine sign language classifiers in front of the camera. The user was asked to place their hand in front of the noise-free background, as in figure 4. Results were shown to the user, in real-time, by the writing above the hand as in Figure 5. The video was manually observed during the test for results. Once it had been manually established that the correct sign was made, this was compared to the output of the system. A

percentage was determined and the results obtained were as follows:

HAND GESTURE	RECOGNITION RATE
Palm/Flat	100%
Two Long Thin Bent Extensions	100%
Narrow/Shallow flat object	100%
Round/Spherical Object	50%
Long Flat Smooth Surface	100%
Fist	50%
Flat/Triangular Object	100%
Index Five	100%
Compact Mass with Salient Extension	100%

TABLE I
TEST DATA OBTAINED DURING TESTING

Overall accuracy of the system reached 88.9%. The system was accurate in identifying all but 2 of the sign classifiers. There was ambiguity in the system when identifying the "fist" and "Round/Spherical Object" hand classifiers. These classifiers can be seen in figure 5 as they were identified during testing. It is clear that these hand classifiers are very similar in shape and orientation. This is the likely reason for the confusion between the two.

VII. CONCLUSION

A system was successfully developed to extend the OpenCV eigenvector functionality to accept and recognize real time video of human hands. The system has performed with an acceptable level of accuracy for the trained hand shapes.

VIII. FUTURE WORK

Tracking, scale and rotation remain a problem in recognition. The system requires that the user conform to restrictions placed for accurate recognition. A greater variety of training subjects would also assist recognition.

Methods of tracking the hand which do not impact the speed of the program would be beneficial.

REFERENCES

[1] Hase, H. and Shinokawa, T. and Yoneda, M. and Suen, C.Y. Recognition of rotated characters by Eigen-space. 2003 .
[2] Cooper, H. and Bowden, R. Large Lexicon Detection of Sign Language. 2007.
[3] Leonardis, A. and Bischof, H. Dealing with Occlusions in the Eigenspace Approach .1996.
[4] Kirby, M. and Sirovich, L. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. 1990 .PAMI. Vol. 12.
[5] Sirovich, L. and Kirby, M. Low Dimensional Procedure for the Characterization of Human Faces. 1987 . JOSA-A . VOLUME 4.
[6] Pentland, A.P. and Moghaddam, B. and Starner, T.E. View-Based and Modular Eigenspaces for Face Recognition", . 1994 . 84-91.
[7] Raphael Cendrillon and Brian C. Lovell. Real-Time Face Recognition using Eigenfaces
[8] Black, M.J. and Jepson, A.D. Eigentracking: Robust Matching And Tracking Of Articulated Objects Using A View-Based Representation . 1998. IJCV . Vol. 26.
[9] Pentland, A.P. and Weaver, J. and Starner, T.E. 1998. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video.
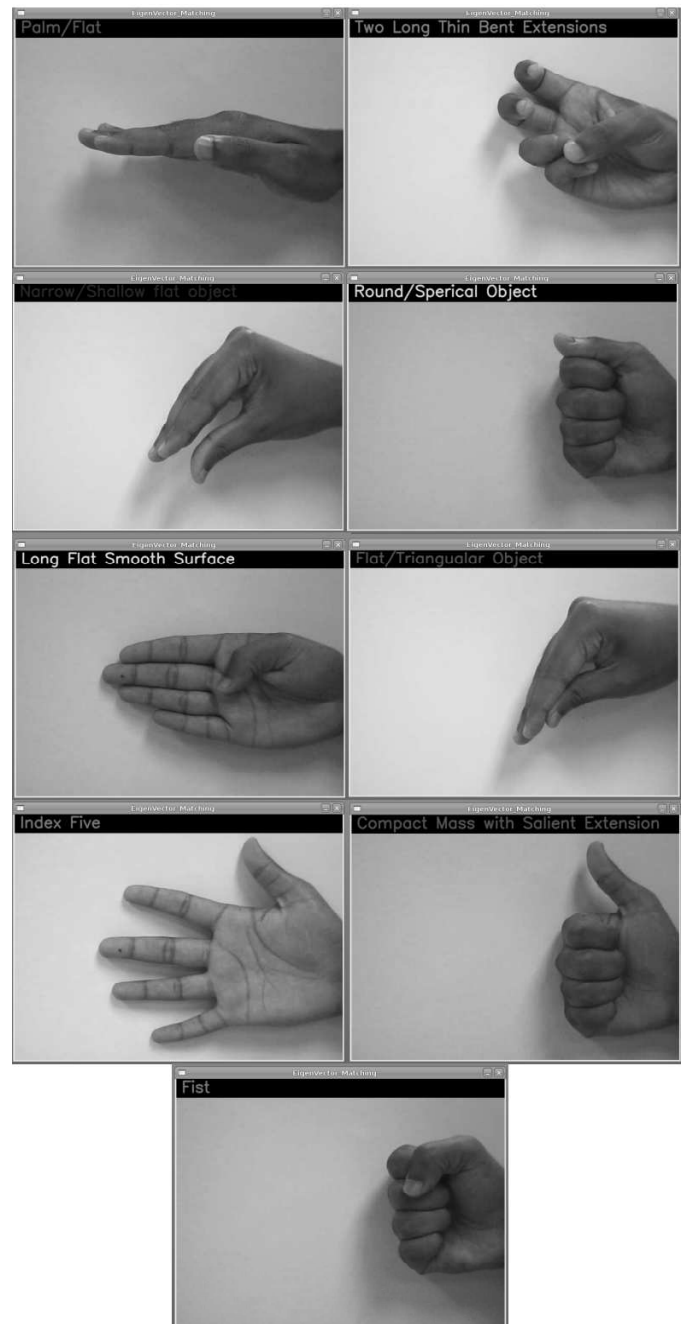
Fig. 5. Examples of positive results shown during testing

Vaughn Segers is currently a Telkom Centre of Excellence M.Sc student at the University of the Western Cape. His area of research is focused on applications for the Deaf and hard of hearing.

James Connan heads up the South African Sign Language (SASL) research group. He has a wide range of interests that include: databases, computer vision and machine learning.