Fast Face Recognition

Karl B. J. Axnick[1] and Kim C. Ng[1]

[1] Intelligent Robotics Research Centre (IRRC), ARC Centre for Perceptive and Intelligent Machines in Complex Environments (PIMCE) Monash University, Melbourne, Australia.

Email: {karl.axnick : Kim.C.Ng}@eng.monash.edu.au

Abstract

This paper introduces an algorithm for face recognition that is fast, robust and accurate. It is designed primarily for access control applications involving small databases such as access to a building, a laboratory or equipment. The algorithm is robust enough to handle inputs from varying sources (2D, 3D and infrared) to detect and recognise faces quickly even when those faces are varied from the database images with pose, illumination and expression changes. It uses common image processing techniques and heuristics to rapidly find salient feature points on a face. Normalised geometric distances and angles are extracted from these salient point locations to generate a unique signature for the face in the image. The performance of the final system has been tested and it achieves recognition speeds of less than 1 second per face at accuracies from 73.5% to 100% depending on the input image type.

Keywords: Face Recognition, Face Detection, Salient Feature Points

1 Introduction

Face recognition is a huge research area [1] and each year the attempted solutions grow in complexity and execution times [2]. Although the complexity improves the methods' accuracies, the achieved accuracy is still not good enough for the "Mecca" of face recognition which is accurate crowd surveillance and global identity recognition. There are two main approaches for face recognition, holistic and geometric [3]. Geometric approaches dominated in the 1980's where simple measurements such as the distance between the eyes and shapes of lines connecting facial features [4] were used to recognise faces, while holistic methods became very popular in the 1990's with the well known approach of Eigenfaces [5]. Even though holistic methods such as neural networks [2] are more complex to implement than their geometric counterparts, their application is much more straight forward, whereby an entire image segment can be reduced to a few key values for comparison with other stored key values and no exact measures or knowledge such as eye locations or the presence of moustaches needs to be known. The problem with this "grab all" approach was that noise, occlusions such as glasses and any other non-face image attribute could be learned by the holistic algorithm and become part of the recognition result even though such factors are not unique to faces.

The new millennium saw the advent of an amalgam between the two approaches whereby holistic techniques (such as Gabor filters [6]) were applied locally around salient feature points (a geometric technique). Although this new paradigm allows many non-unique features in the image to be ignored and the accuracy is very good (96.7% [6]), local non-unique attributes can still sneak through and the time to run such algorithms on large databases is unwieldy (3 seconds per possible face [7]). Complexity is a necessity to differentiate between faces in a large database. For access control applications involving small databases (of 100 people at most) simple, fast and accurate techniques are desirable. The algorithm proposed in this paper allows for and have achieved the actual implementation of face recognition into current systems without further delay.

Although many face recognition algorithms have already been very successful with small databases [4, 6, and 7], they were not aiming to solve the small database recognition problem. Instead the small database was a test bed for estimating large database performance. By aiming specifically at smaller databases this paper's method achieves better results than those methods listed.

This paper introduces its new technique by first explaining the methods used for face detection in Section 2, followed by the feature finding algorithms in Section 3. Finally Section 4 explains how face recognition is achieved. Some experimental results are listed and explained in Section 5 and Section 6 draws the conclusions.

2 Face Detection/Localisation

Face detection is a bottleneck for any face recognition system where the target is not held in a controlled state for scanning (crowd surveillance for example as opposed to in front of an ATM). Viola and Jones [8] overcame this bottleneck by using a combination of many weak filters to quickly capture possible faces in the image. A more modern version of this method was used in this research (OpenCV's Haar Face Detector[1] Lienhart [9]) to verify a novel face detection that uses background subtraction, blob analysis and eye recognition. Since this method is faster and more robust than Lienhart's method and has comparable accuracy it was used in the actual system implementation and speed tests. Lienhart's method was also used, but for checking the paper's algorithm accuracy as the database composed of still images and random backgrounds, making the background subtraction approach inadmissible in some tests.

[1] http://www.intel.com/research/mrl/research/opencv/

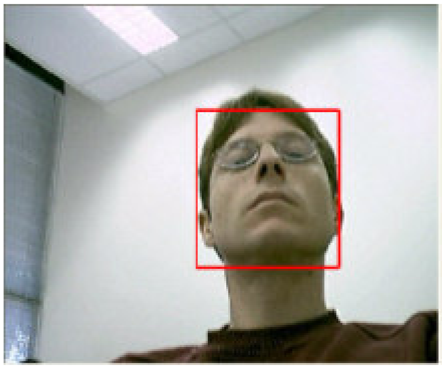


Fig. 1: The result of OpenCV's Face Detector

2.1 Background Subtraction

Background subtraction is very common in video processing and this paper uses a method similar to [10]. Once movement is detected in the face recognition system's view, segmented binary motion blobs are created. Each blob is assumed to be a possible face and a 'maybe face' boundary region is drawn over the blob at the top of the blob's mass. The concaved edges around the neck area delineate the bottom boundary of this face region. A secondary heuristic that estimates the head height based on the head width is used when the primary heuristic that uses the neck's concavity fails because the neck is occluded by long hair or clothing. Fig. 2a and 2b show the 'maybe face' regions found on blobs from both colour and thermal 2D video respectively. Two assumptions are made that will not critically harm the final result if proved wrong. These are that all moving objects in a video sequence are humans and that the faces of humans are contained in the top part of the movement blobs.

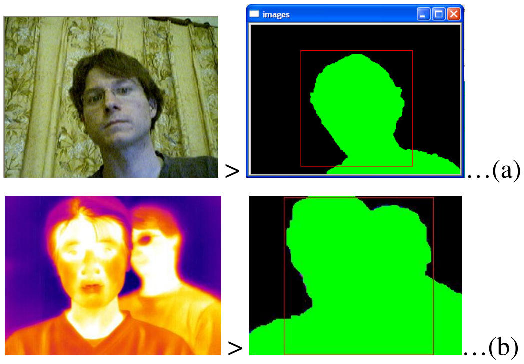


Fig. 2: Finding the Face Boundary Boxes a) from colour video, b) using infrared video (the heads are too close and so the 'maybe face' region covers both as one).

The 'maybe face' boundary region is purposely drawn larger than the motion blob's head to maximise the chance of having a face completely inside the 'maybe face' boundary. The disadvantage is that it may include data from multiple faces if the people are crossing paths or are moving close together (Fig. 2b). Section 2.2 resolves this difficulty by choosing only the two most visible and similar eyes (which on average will be from the one face). Of course this means that the other face will be missed. This is not of major concern with video inputs as subsequent frames will most likely present the subject's face again and probably on its own.

2.2 Eye Localisation

After Section 2.1 returns a 'maybe face' image segment, the algorithm then determines if a face is present in the image and if so where are the eyes? It then uses these eye locations to create a more accurate bounding box for the face.

Before any processing is done we must normalise the 'maybe face' space. This involves a convolution with a Gaussian Kernel (1), followed by a contrast stretching operation (2).

$$G(x,y) = \tfrac{1}{2}\pi\delta^2.\exp\left\{\frac{x^2+y^2}{2\delta^2}\right\} \qquad \text{……… (1)}$$

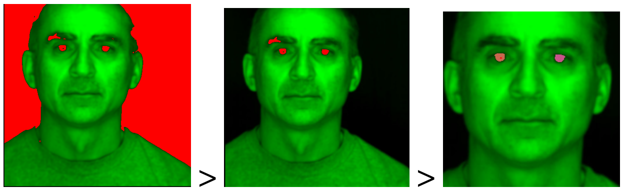
$$\begin{array}{lll} \text{if} & a[m,n] \le p_{low} & \text{then } b[m,n] = 0 \\ \text{if} & p_{low} < a[m,n] < p_{high} & \text{then } b[m,n] = \dfrac{255a[m,n] - p_{low}}{p_{high} - p_{low}} \\ \text{if} & \mathrm{a[m,n]} \ge \mathrm{p}_{\mathrm{high}} & \text{then } \mathrm{b[m,n]} = 255 \end{array} \qquad \text{……… (2)}$$

where a[.] is the input image and b[.] is the output.

After normalisation two new filters are then used. One is a Laplacian filter that finds sharp contrast changes in a circle (finds red eye effect or sharp eye reflections in the image, and pupils if clearly visible), the other is simply a binary threshold that filters out pixels with intensity values greater than the lowest 5% of intensity values in the image histogram (finds the dark pupils).

These two filters find many points on an average image and so the results need to be consolidated and rapidly filtered. This is done by first dilating the binary blobs so that small blobs in close proximity will merge, followed by several erosions to remove small noise blobs that have not merged. Finally the remaining blobs are labelled and have their statistical parameters found (central moments, area, second order moments etc.). These parameters are also analysed and have heuristic filters applied to remove more obvious non-eyes. These include properties such as being long and thin blobs (non-circular).

Next we check if any combinations of those suspected eyes make sense. If all combinations of one particular eye location with all the other possible eyes cannot make an angle of less than 30 degrees, and its radius differ by more than 20% from all the other eyes then that eye cannot be a real eye as it does not have a viable partner. Of course a profile view could invalidate this assumption, but we are only looking for faces where both eyes are visible, so around 45-60 degrees off centre is the maximum allowable pose angle for this paper's method. Also valid eye pairs must be within a certain distance to each other relative to their diameter sizes. If these filtering processes result in there being only one or zero possible eyes left in the image then the image segment is discarded as having no face. Fig. 3 shows the face and eye localisation process.



(a) (b) (c)

Fig. 3: Eye Localisation. a) all dark pixels found, b) remove obvious non eyes, c) only eyes with viable partners left.

3 Feature Extraction

At this stage there are at least two possible eyes in the supplied image segment. However, only some pairings of these possible eyes are allowable which means there are only a handful of possible face pose assumptions. The following sub-sections find the most probable pose based on the possible eyes and features relative to the salient knowledge of where the eyes are (for instance the nose is below the eyes and the mouth below the nose).

3.1 Finding the Eye Parameters

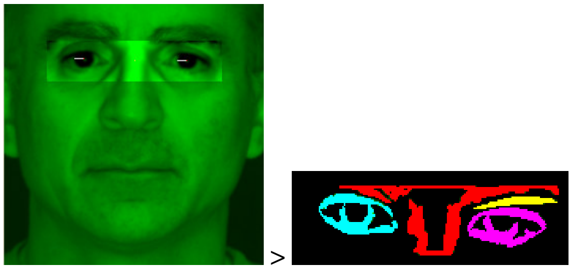
For each possible eye location we start with a minimum possible pupil radius and iteratively apply equation (3) from that location. The aim is to find an x, y pair and a pupil radius that maximises τ (a ratio of contrast between circles of differing radii). If the pupil's radius grows too large or the optimal eye x, y location moves too far from the original possible eye start point then the current possible eye is rejected as a real eye. If the translations convolve with the centre point of another possible eye point then that other possible eye is removed as a possible eye, but the current possible eye is allowed to translate further before being invalidated.

$$\tau = \frac{\left\{\frac{sum(\underline{x},\underline{y},\underline{r}+1)-sum(\underline{x},\underline{y},\underline{r})}{\underline{r}+1}\right\}}{\left\{\frac{sum(\underline{x},\underline{y},\underline{r})-sum(\underline{x},\underline{y},\underline{r}-1)}{\underline{r}}\right\}} \qquad \text{.........(3)}$$

where sum(.) simply sums the pixels in the circle within the input image defined by the arguments.

Once τ has been maximised for all of the remaining possible eyes, a repeat of the above algorithm checks that all possible eye pairs have similar radius values and that the inter eye distances are valid compared to the pupil radii etc. The eye pair with the highest τ sum is classified as the real eye pair, with the eye centres and pupil radii recorded.

Next we find the eye outers and inners. We normalise only the local area around each eye using (1) and (2). Then a Sobel operator is convolved on this small space to find the eye outline (See Fig. 4b). Blob analysis of the outline structure quickly reveals the extreme top and bottom values (the top and bottom of the eye) and the extreme left and right values (the eye inner and outer, depending on which eye is currently being examined).



(a) (b)

Fig. 4: Finding the Eye Parameters. a) normalise locally, b) use edge detection and blob analysis

3.2 Finding the Nose Parameters

The nose point will be perpendicular from the centre of a line connecting both eye centres (the perpendicular bisector). From biology [11] the most probable range to find the nose point of a naturally formed nose along the perpendicular bisector line can be calculated from the inter eye distance and pupil radii. By normalising this most probable nose area locally and using a Sobel operator to get the nose edges the nose point is found (it is the lowest point on the eye bisector line that convolves with the nose outline blob). We then use blob analysis on the blob that contains the nose point (see Fig. 5). The extreme left and right values for that blob give the nose outers' locations. Note however, that when we refer to the extreme top, bottom, left and right blob values as in this and the previous section, the axis of such observations are not the standard ones innate to the image but to the innate axis rotated and translated relevant to the angle between the eye centres and the central point between the eyes.

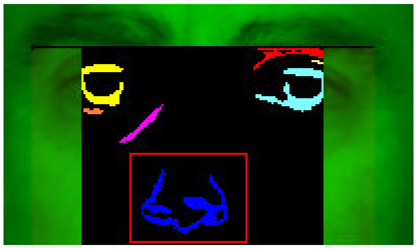


Fig. 5: Blob Containing the Nose Parameters

3.3 Finding the Mouth Parameters

We can calculate an accurate bounding box that will contain the entire mouth without too much non-mouth data Normalisation and a Sobel operation is then applied to the target area.

By following the perpendicular bisector from between the eyes, from the direction of the nose point, the top lip will be encountered and is recorded as the top point of the lips. A blob analysis of the area containing the top lip can then discover several lip properties (Fig. 6). A large solid symmetrical (about the horizontal axis) blob of the lip line tells us that the lips (both top and bottom) are pursed together, making it simple to then find the mouth centre, bottom lip and lip outer salient points. If the lip blob is large and symmetric but has a hollow in the centre, we know that the mouth is open, and we can find the same key points. If the lip blob is non-symmetrical then we know we just have the top lip contained in the current blob and that the mouth is most likely open. By following the perpendicular eye bisector from the bottom centre of the top lip blob, the bottom lip blob can be found. By artificially merging these blobs, the resultant blob can give us the mouth parameters for the final case.

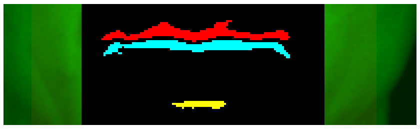


Fig. 6: Blobs Containing the Mouth Parameters

4 Face Recognition

For rapid but accurate face recognition performance in an access control application, the recognition method needs to be robust to noise, pose, illumination and expression (PIE) variations, in order to allow it to be novel and to compete with the other modern face recognition methods [12], [13] which can't handle all of these variations simultaneously [14].

As pose variations are rigid body transformations, the relative distances and angles between the salient points are unaffected by pose variations in 3-Dimensions. However the 2D methods of black-and-white, colour and thermal images will suffer under pose changes. To solve this problem a multiple pose database is needed for the 2D images as was the solution in [15]. This adds a cost and complexity to the algorithm that slows the speed but the accuracy and robustness pay off is worth the cost.

Illumination variations are not problematic as apart from the initial scan to find all possible eyes, all algorithms work on normalised local data The thresholds for finding lines in the different local areas are also customised based on the local histograms. This means that global shading variations will not affect recognition performance.

Expression variations are combated by giving salient points diminished weights when they lie in large expression variation areas, most notably the mouth and other locations learned by differentiating the 3D and 2D scans of many test faces with greatly varying expressions, but with absolutely no other variants.

Noise variation is handled in the following manner: visible points that would have been found but are not found due to corrupted scan data, do not alter the recognition score either positively or negatively. Points that are not lost due to noise but simply contain noise have their effects diminished by using many salient points for recognition. So as long as the noise is not global and large, the effects of noise will be diminished with the following face recognition algorithm.

Scale variation was handled by the normalisation process of the recognition algorithm which follows:

1) Find the Euclidean point distance between all found salient feature points and create an array of these values. Substitute a zero for the distance if a point has not been found.

2) Generate another array which is the result of all Euclidean points' distances from step 1 being divided by each other salient point pair distances in order. Divisions by zero should be detected and a result of zero substituted.

3) Find the Euclidean distance between the normalised Euclidean distance arrays of step 2 with each of the normalised distance arrays stored in the database that have the same pose and expression as the target image. Decrease the distance value involving found points that are vulnerable to expression variation by a learned amount. Increase the found distance value involving points that have been learned to be of high value for recognition accuracy (allow more accurate class classifications).

4) The database face that both has the smallest distance to the current target's normalised distance array, and meets the predicate that the distance is below some threshold is determined to be that target's match.

The above algorithm is very straight forward and as a result quite fast and designed to be robust to all of the possible sources of error. The important first step is finding both eyes. However unlike [16], which used the distance between the eyes to normalise the data, thereby allowing anyone wearing glasses or blinking to break the system, because this paper's system is real-time and processing video, the target would need to be wearing sunglasses or be infinitely blinking to break the system, so the eye finding dependence is not critical. These aversive acts are unlikely to occur as such behaviours would draw attention to the people in question (even if they are not recognised) and as the recognition system prevents access to unauthorised people, being unrecognisable is of no assistance.

5 Experimental Results

As the aim of this paper was to achieve good fast face recognition for access control with a small database, both speed and accuracy need to be analysed within the experiment. Also the algorithm needed to be robust to handle multiple input formats. Finally to be novel the system also has to handle variations in pose, illumination and expression. The database used was created by the authors as accurate 3D scans were required that are not available in free online databases.

Faces were selected at random from a large population on the university campus by asking for volunteers to undergo a 30 minute scan. Both 2D colour and 3D scans were taken simultaneously and these include a range of variations due to pose, illumination and expression. This collected small database is very hard for face recognition due to its huge variations and also its 3 related volunteers. The black and white images were gathered from a previous experiment that also used volunteers. Finally the infrared images came from a research trip (to Prof. Terry Caelli at NICTA, Australia National University, Canberra) that investigated the feasibility of using infrared in face recognition and also contains random volunteers.

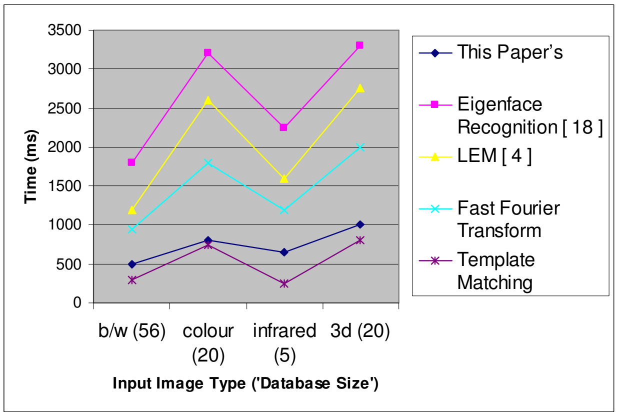


Fig. 7: Recognition Time Comparisons for Different Methods

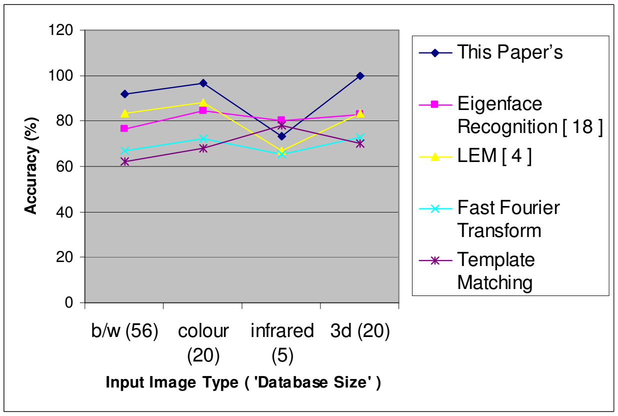


Fig. 8: Face Recognition Accuracy Comparisons for Different Methods

Of the many known and valid successful small database face recognition systems only Eigenfaces and LEM were used as yardsticks for this paper's algorithm because they represent the best and the fastest of these. The two other methods used, Fast Fourier transform and template matching are simple low level methods we created ourselves that can be accurate and are extremely fast.

The well known EigenFace recognition and Line Edge Map (LEM) [4] face recognition methods were implemented without much change from their respective papers' for the black-and-white, colour and thermal 2D images. However for the 3D images, rather than R,G and B being used, the normalised X,Y and Z normals of the face vertices were fed to the face recognition methods in 2D arrays (e.g. a 2D colour image of the face's normals).

Fast Fourier transform face recognition is similar to EigenFace recognition, except that rather than finding the greatest variations along the greatest variation axis, only the greatest variation frequencies are found with a static axis. Only values from the top 40 most variant frequencies (as identified from experimentation) are employed as values in the converted faces recognition signature.

The template matching approach is the simplest and fastest of all the methods. It is employed by first aligning the eyes of the target image with the eyes of the current database image being checked (the database image is normalised to have the same inter-eye distance of the target image, so a perfect alignment is possible) and then a sum of aligned pixel differences for the entire face space is returned as a match score. Obviously the lower the match score the higher the chance that the faces are a match, for all input types.

The experiments are to compare the face recognition performances. Therefore all of the methods used the exact same initial algorithms (as listed in this paper) up to and including Section 3.1. This means that if the eyes are not correctly identified then all the algorithms will fail. Once two real eyes were found in an image, all five of the methods being tested were sent an identical image segment (a normalised and scaled (on eye length) image (containing only face data) of a set size (256x256), from Section 3.1 of our algorithm) that was guaranteed to contain all of the facial features and all of the face, with no background or hats or earrings etc.

The Fig. 7 recognition time comparisons show only two algorithms as having a sub one second recognition performance, this paper's and template matching. However a quick comparison of accuracy in Fig. 8 shows that the template matching recognition accuracy is quite poor. The template accuracy could be improved using local templates but that would increase its run time. The popular EigenFace method displayed good accuracy results but as expected was the slowest in recognition speed. LEM had comparable accuracy to this paper's but was corrupted with PIE and noise variations in the database images as it used more non-rigid points than our method.

The accuracy is also dependent on the type of database. As both the colour 2D images and the 3D images had been collected simultaneously the comparison between their results is more meaningful. 3D images with their extra data allow for far more accurate face recognition (consistently 100%) than their 2D image counterparts. This trend of extra data aiding recognition is also evident when comparing the monochrome and colour results. This trend is non-linear however, as holistic techniques which use far more data than the 3D method achieve less recognition accuracy due to their increased susceptibility to variations within the data caused by noise and PIE variations.

Results were poor for all methods on the infrared input even though the database had only 5 people. This was due to the poor spatial quality of the images and both EigenFace and template matching top scored showing the advantage of holistic approaches.

6 Conclusion

We have demonstrated a fast face recognition system that is quite accurate even though pose, illumination and expression variations were present. The performance with 2D colour images was 97% accuracy with one second processing time per face and this system has been tested in real time on ***live*** faces for access control successfully. As more scans are added to the system the accuracy will likely fall, however. The use of locally focussed neural networks around the salient feature points (which will be mapped to hardware) is currently being investigated. The 3D method has already achieved 100% accuracy and is readily deployable for use in access control. Unfortunately the current 3D scans require that the subject stay still for

0.4 seconds while an eye safe laser scans them. A more user friendly and more covert stereo vision system is being developed to speed up scanning and improve throughput. However as stereo vision will not be as accurate and robust as the laser, a performance decrease might result.

In contrast, holistic face recognition methods which use entire face images in a 'grab all' manner, can easily be corrupted through the pose, illumination and expression variations [17]. Other geometric methods such as local Gabor Wavelet filters [6] would however yield more accurate results than this paper's geometric method. Wavelet examination of the local area data around salient points is more detailed and robust (as it tests a lot more points) compared to the heuristic methods listed in Section 3. However the increased complexity and running time of such methods may rule them out for access control applications. Also by using the local areas as recognisable features these algorithms are less resistant to expression, shade and pose variations.

References

[1] W. Zhao, R. Chellappa, and A. Rosenfeld, "Face recognition: a literature survey". *ACM Computing Surveys*, Vol. 35:pp. 399–458, December 2003.

[2] J.E. Meng, W. Chen and W. Shiqian, "High-speed face recognition based on discrete cosine transform and RBF neural networks"*; IEEE Transactions on Neural Networks,* Vol. 16, Issue 3, Page(s):679 – 691, May 2005.

[3] V. Bruce, P.J.B. Hancock and A.M. Burton, "Comparisons between human and computer recognition of faces", *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Vol., Iss., 14-16 Pages:408-413, Apr 1998

[4] G. Yongsheng and M.K.H. Leung, "Face recognition using line edge map", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , Vol.24, Iss.6, Pages:764-779, Jun 2002

[5] M. Turk and A. Pentland. "Eigenfaces for recognition". *Journal of Cognitive Neuroscience*, 3, 71-86, 1991

[6] S. Da-Rui and W. Le-Nan, "A local-to-holistic face recognition approach using elastic graph matching" *Proc. International Conference on Machine Learning and Cybernetics,* Vol. 1, Page(s):240 - 242, 4-5 Nov. 2002.

[7] W. Haiyuan, Y. Yoshida, T. Shioyama, "Optimal Gabor filters for high speed face identification" *Proc. 16th International Conference on Pattern Recognition,* Vol. 1, pp.:107 - 110, 11-15 Aug 2002.

[8] 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*, CVPR 2001.

[9] R. Lienhart and J. Maydt, "An extended Set of Haar-like Features for Rapid Object Detection", *IEEE ICIP 2002*, Vol 1, pp. 900-903, Sep.2002

[10] Z. Qi, R. Klette, "Robust background subtraction and maintenance", *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004.* Vol. 2, pp.:90 - 93, 23-26 Aug. 2004.

[11] V. Bruce and A. Young, "Understanding face recognition". *The British Journal of Psychology*, 77 (3), 305-327, 1986.

[12] J. Ruiz-del-Solar and P. Navarrete, "Eigenspace-based face recognition: a comparative study of different approaches", *IEEE Transactions on Systems, Man and Cybernetics, Part C,* Vol. 35, Issue 3, Page(s):315 – 325, Aug. 2005.

[13] K.I. Chang, K.W. Bowyer and P.J. Flynn, "An evaluation of multimodal 2D+3D face biometrics", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 27, Issue 4, pp.:619 – 624, April 2005.

[14] Y. Hu et al., "Automatic 3D reconstruction for face recognition". *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004*.

[15] A.Pentland, B. Moghaddam, T. Starner, O. Oliyide, and M. Turk., "View-Based and Modular Eigenspaces for Face Recognition", *Technical Report 245, MIT Media Lab*, 1993.

[16] H. Weimin and R. Mariani, "Face detection and precise eyes location", *Proceedings 15th International Conference on Pattern Recognition,* Vol. 4, pp.:722 – 727, 3-7 Sept. 2000.

[17] J. Lai, P. Yuen, G. Feng, "Face Recognition Using Holistic Fourier Invariant Features", *Pattern Recognition*, 34(1), pp95-109, 2001.

[18] T.J. Chin and D. Suter, MECSE-6-2004: "A Study of the Eigenface Approach for Face Recognition", IRRC, ECSE, *Monash University,* June 2004.