A Real-Time Face Recognition System Using Custom VLSI Hardware

Jeffrey M. Gilbert and Woodward Yang

Harvard University, Cambridge, MA 02138

Tel (617) 495-3987 Fax (617) 495-9837

Abstract

A real-time face recognition system has been implemented on an IBM compatible personal computer with a video camera, image digitizer, and custom VLSI image correlator chip. With a single frontal facial image under semi-controlled lighting conditions, the system performs (i) image preprocessing and template extraction, (ii) template correlation with a database of 173 images, and (iii) postprocessing of correlation results to identify the user. System performance issues including image preprocessing, face recognition algorithm, software development, and VLSI hardware implementation are addressed. In particular, the parallel, fully pipelined VLSI image correlator is able to perform 340 Mop/second and achieve a speed up of 20 over optimized assembly code on a 80486/66DX2. The complete system is able to identify a user from a database of 173 images of 34 persons in approximately 2 to 3 seconds. While the recognition performance of the system is difficult to quantify simply, the system achieves a very conservative 88% recognition rate using cross-validation on the moderately varied database.

I. Introduction

Humans are able to recognize faces effortlessly under all kinds of adverse conditions, but this simple task has been difficult for computer systems even under fairly constrained conditions. Successful face recognition entails the ability to identify the same person under different circumstances while distinguishing between individuals. Variations in scale, position, illumination, orientation, and facial expression make it difficult to distinguish the intrinsic differences between two different faces while ignoring differences caused by the environment. Even when acceptable recognition has been accomplished with a computer, the actual implementation has typically required long run times on high performance workstations or the use of expensive supercomputers. The goal of this work is to develop an efficient, real-time face recognition system that would be able to recognize a person in a matter of a few seconds [4].

Face recognition has been the focus of computer vision researchers for many years. There are two basic approaches to face recognition, (i) parameter-based and (ii) template-based. In parameter-based recognition, the facial image is analyzed and reduced to a small number of parameters describing important facial features such as the eye shape, nose location, and cheek bone curvature. These few extracted facial parameters are subsequently compared to database of known faces. Parameter-based recognition schemes attempt to develop an efficient representation of salient features of an individual [1]. While the database search and comparison for parameter-based recognition may not be computationally intensive, the image processing required to extract the appropriate parameters is quite computationally expensive and requires careful selection of facial parameters which will unambiguously describe an individual's face.

In template-based recognition, the salient regions of the facial image are extracted and directly compared on a pixel-by-pixel basis with an image database of known individuals. While the image preprocessing can be much simpler, the database search and comparison for template-based recognition can be quite computationally expensive since image regions rather than single parameters are being compared. In particular, this work uses a template-based recognition scheme that was originally outlined by Brunelli and Poggio [2]. Initially, the user presents a frontal facial image, and the system automatically attempts to find the eye locations. The eye locations are used to guide the extraction of four templates corresponding to the overall face, eyes, nose, and mouth region. These four regions are subsequently correlated with an image database, and the highest total correlation is used to identify the user. While Poggio and Brunelli are able to achieve a remarkable 100% recognition rate over a database of 188 images of 47 persons, their system, which runs on a high performance workstation, is not able to run in real-time.

The applications for a face recognition system range from simple security to intelligent user interfaces. While physical keys and secret passwords are the most common and conventional methods for identification of individuals, they impose an obvious burden on users and are susceptible to fraud. In contrast, biometric systems attempt to identify persons by utilizing inherent physical features of humans such as finger prints, retinal patterns, and vocal characteristics. Effective biometric identification systems should be easy to use and less susceptible to fraud. In particular, facial features are an obvious and effective biometric of individuals, and the ability to recognize individuals from their faces is an integral part of human society. While any computer (or human) face recognition system has obvious limitations such as identical twins or masks, face recognition could be used in combination with other biometric or security systems to provide a much higher level of security surpassing that of any individual system. However, the primary advantages of face recognition is likely to be its non-invasive nature and socially acceptable method for identifying individuals especially when compared with finger print analysis or retinal scanning.

II. **Face Recognition Task**

The face recognition system was based in large part on a template-based face recognition algorithm described by Brunelli and Poggio [2]. The actual recognition process can be broken down into three distinct phases (see Figure 1): (i) image preprocessing and template extraction and normalization, (ii) template correlation with image database, and (iii) postprocessing of correlation scores to identify user with high confidence. From a single frontal facial image under semi-controlled lighting conditions and limited number of facial expressions, the system can robustly identify a user from an image database of 173 images of 34 persons. While the recognition performance of the system is difficult to quantify simply, the system achieves a very conservative 88% recognition rate using cross-validation on the moderately varied database.

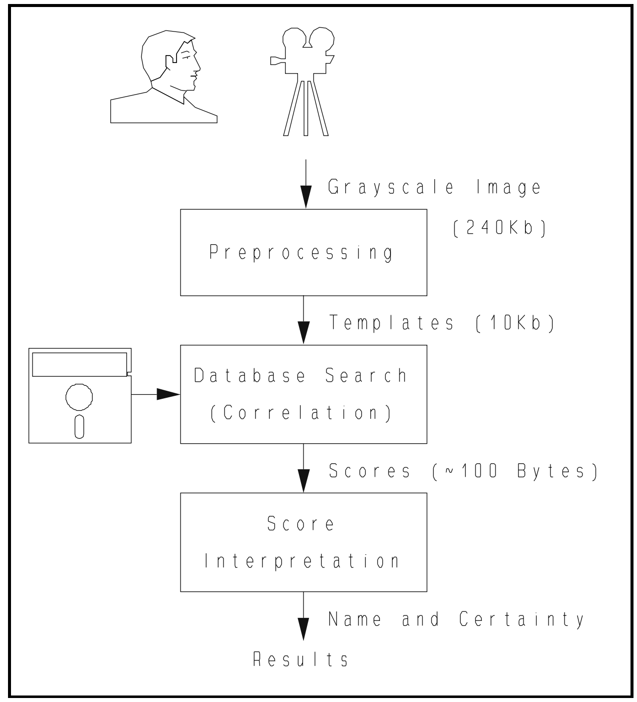

Figure 1 Overall Processing Data Flow

Image Preprocessing

Image preprocessing entails transforming a 512x480 grey-level image into four intensity normalized templates corresponding to the eyes, nose, mouth, and the entire face (excluding hair, ears etc.) of the user. The regions of the image corresponding to the templates are located by finding the user's eyes and normalizing the image scale based on the eye positions and inter-ocular distance.

Eye Location

Locating eyes in a visually complex image in real-time is a formidable task. The goal of the real-time face recognition system is to operate in such a manner as to minimally constrain the user's position within the image. This requires the ability to find the eyes at varying scales over a range of locations in the image. Since the accuracy of the eye location affects the extraction of the templates, and thus the correlation and recognition, the location process must be precise. The location process is

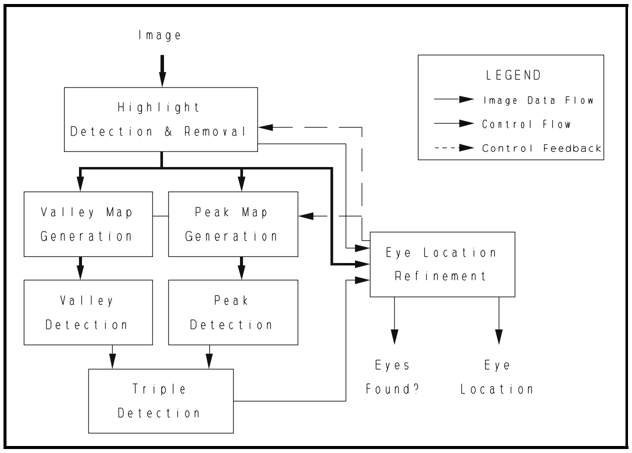

Figure 2 Block Diagram of Eye Location Algorithm

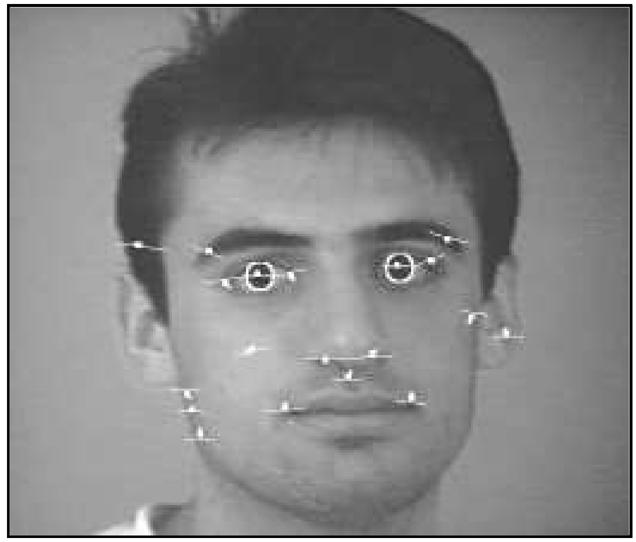

Figure 3 Results of Automatic Eye Location showing Rough (hash marks) and Refined (circles) Eye Locations

divided into two parts - rough location and refinement (See Figure 2). The rough location phase quickly scans the image and generates a list of candidate eye locations. The rough eye location algorithm is based on the observation that an eye is distinguished by the presence of a large dark blob, the iris, surrounded by smaller light blobs on each side, the whites [5]. However, under certain lighting conditions, highlights within the eyes need to be removed and can also be used as additional cues for eye location. When coupled with sufficient high-level constraints on the relative positions of the blobs and an acceptable measure of the "blobbiness", this simple system performs remarkably well. The refinement stage then looks more closely at these areas to determine more exactly the best fit for an eye, given inter-ocular constraints. The refinement process not only assigns a more exact location to each of the candidate eyes, but also assigns a radius to the iris (see Figure 3). This allows more selective pruning by imposing the restriction that the two eyes be of similar size. In addition, the inter-ocular spacing is constrained to a distance proportional to the eye size.

Template Extraction and Normalization

Once the eyes are located, subsampled templates of the face, eyes, nose, and mouth are extracted (see Figure 4). The inter-ocular distance is taken as a scaling factor, and the inter-ocular axis is normalized to be horizontal. The four regions of the image are determined by fixed ratios and offsets relative to the eyes. Skewless affine transformations are used to scale and rotate four area of the image into the four templates. When multiple image pixels correspond to a single template pixel, averaging is employed. The template sizes are fixed but tailored to the size of the region from which they are extracted. The face template is 68×68, the eye template is 68×34, while the nose and mouth templates are each 34×34. The template size governs the accuracy and speed of the database search. Choosing the templates to be too small results in a loss of information. Choosing the templates too large results in the extraction and correlation processes running slowly. In addition, the registration and alignment errors become more severe with larger template sizes.

Once the templates have been extracted, they must be normalized for variations in lighting to ensure accurate

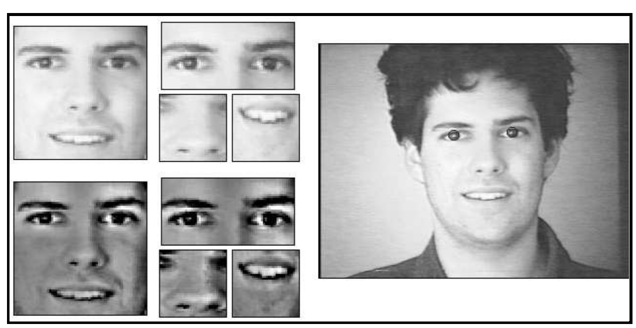

Figure 4 Results of Template Extraction and Normalization

correlation between the templates. If the image intensity is used directly, a dark image of one person could match better with a dark image of a different person than with a light image of the same person. Since the lighting conditions prevailing at the time of the image database creation may be different from those at the time of recognition, insensitivity to lighting conditions is crucial. Two types of template intensity normalization are employed, local normalization and global normalization. Local normalization entails dividing the pixel intensity at a given point by the average intensity in a surrounding neighborhood. This is roughly equivalent to high pass filtering of the template data spatially and removes intensity gradients caused by non-uniform lighting. Global normalization consists of determining the mean and standard deviation of the template and normalizing the pixel values to compensate for low variance due to dim lighting or image saturation.

Template Correlation with Image Database

After the facial image of the user has been preprocessed to obtain the normalized templates, the templates are compared to those in an image database of known persons. Templates are compared to those in the database by a robust correlation process to compensate for possible registration errors. In particular, the template is compared to database images over a range of 25 different alignments corresponding to spatial shifts between +2 and -2 pixels in both the horizontal and vertical directions. In addition, the image correlations are actually computed as the sum-of-absolute-differences for computational efficiency and for more robust matching performance. While absolute-difference correlation is more efficient than multiplication based correlation, it is still a time consuming process. Each set of four templates consists of roughly 10,000 pixels. Thus each template comparison over the 25 different alignments requires approximately 250,000 absolute value and sum operations. An Intel 80486/66DX2 running optimized assembly code can only perform roughly 5 million integer absolute value and sum operations per second including data movement and other overhead. This would seem to limit the database search rate to 20 template sets per second, severely constraining the size of the database possible for real-time operation.

Multi-resolution correlation has been employed to reduce the size and number of correlations. Multi-resolution correlation entails performing an initial set of lower-resolution correlations to limit the database to a smaller number of candidates which could be the unknown individual [3]. The results are not accurate enough to generate a definitive answer, but can be used to narrow the individual's identity to ten candidates in a fraction of the time that a full-resolution search requires. The top ten candidates are then compared at full resolution to the unknown individual to yield the final result. In this way, full accuracy is maintained at only a fraction of the computational time. On a database of 200 images, multi-resolution correlation is responsible for a speedup by a factor of over two.

This still would not result in real-time performance for a reasonably sized database. Thus, a special purpose VLSI image correlator was implemented to accelerate the image database correlation and search, as is described in Section III.

Postprocessing of Correlation Scores

The correlation of the normalized extracted templates from the target image with the database templates generates a list of the top ten candidates and their correlation scores. The task of the postprocessing stage is to interpret the corresponding correlation scores and determine if they indicate a match with someone previously stored in the image database. Typically this is not a clear-cut decision, therefore decisions have an associated measure of confidence. The goal is to recognize as many images as possible while missing and mistakenly recognizing as few images as possible. An image is recognized if the system correctly identifies it as corresponding to someone who is in the database. An image is missed if the user is in the database and the system fails to identify him or her. Finally, an image is mistakenly recognized if the system claims that the user corresponds to a person in the database, and the user is actually a different person in the database or is not represented in the database. Postprocessing attempts to maximize the recognition rate while minimizing the mistaken and mis-recognition rate by interpreting the raw correlation scores with an intelligent and robust decision making process.

Even from a cursory inspection of the raw scores produced by the correlation stage it is evident that the decision to recognize or reject an image is not an obvious one in many cases. Scores from correct matches are sometimes worse than those from false matches. Although the raw scores contain all of the information generated from the correlation stage, additional pseudo-scores are derived to aid in the process of score interpretation and identification. A composite score is computed as the weighted sum of the four independent template (face, eye, nose, and mouth) scores.

Additional pseudo-scores are also computed by examining the relationship between the raw correlation scores of the top ten candidates (see Figure 5). This is to account for global variations in the correlations scores due variations in lighting and pose. For example, if an input image was acquired in conditions which differed greatly from the conditions under which the database templates were captured, then all of the correlation scores will be worse (higher sum of absolute differences). However, the correct database templates may still match comparatively well. To exploit this observation, each of the top ten candidates is compared to a "reference" candidate. The top candidate is compared to the second ranking candidate. The second through tenth candidates are compared to the top ranking candidate. The comparisons take the form of computing a ratio and difference between each candidate and its reference candidate. Thus for each of the five templates (face, eyes, nose, mouth, and composite) of the top ten candidates, a ratio and difference is computed, increasing the total number of scores per candidate to 15.

The 15 correlation scores and pseudo-scores for each of the ten candidates must then be interpreted to determine which, if any, of the candidates match the input image.

An adaptive, robust decision making process is implemented by assigning global threshold values for acceptance and rejection of each of the raw correlation scores and pseudo-scores. In particular, any candidate with match scores worse than the rejection threshold is immediately rejected. However, positive identification of the user is only possible if a single candidate has match scores above the acceptance threshold. Otherwise the user is rejected as an unidentified individual. Match scores that fall between the rejection and acceptance are interpreted as being ambiguous and are ignored. This simple postprocessing scheme provides a robust interpretation for identifying individuals from raw template correlation scores and their derived pseudo-scores. Since the acceptance and rejection thresholds depend on the environment and the characteristics of individuals in the database, it is desirable to have the system "learn" the appropriate values. In particular, the database is directly used to determine the appropriate thresholds for acceptance and rejection through cross-validation.

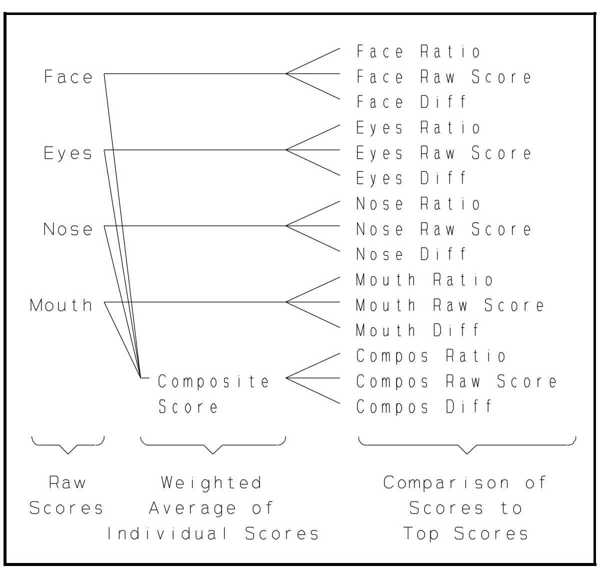

Figure 5 Correlation Pseudo-Scores

III. **System Architecture**

The system hardware consists of an IBM PC 80486/DX2, a commercial frame grabber, video camera, and custom VLSI hardware (see Figure 6). The goal of the hardware system architecture is to extract the highest performance from these components.

Software implementation of the face recognition system described above on an IBM PC will be limited by a computational bottleneck associated with the image database correlation. Benchmarks on an Intel 80486/66DX2 system (see Table I) reveal that real-time

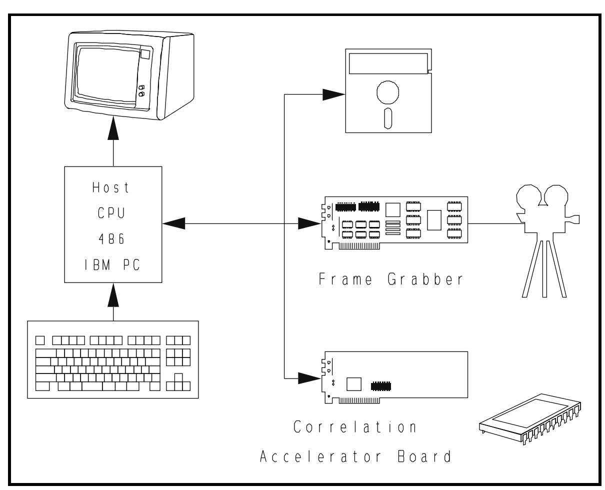

Figure 6 Hardware System Overview

performance in software alone would not be possible with a moderately sized database of 500 images. Thus, in order to achieve real-time performance, a special purpose VLSI image correlator was implemented and integrated into the system as a coprocessor board on the ISA bus.

	One 68×68 Template	500 Sets Brute Force
486/DX2 IBM PC:		
Naive C Code	140 ms	140 sec
Better C Code	60 ms	60 sec
Assembly	20 ms	20 sec
VLSI Hardware measured	0.9 ms	0.9 sec
VLSI Hardware theoretical	0.6 ms	0.6 sec
VLSI & DSP estim.	~0.25 ms	~0.25 sec
Sun SparcStation:		
Naive C Code	77 ms	77 sec
Better C Code	56 ms	56 sec
DECstation 3100:		
Naive C Code	93 ms	93 sec
Better C Code	67 ms	67 sec
DECstation 5400:		
Naive C Code	79 ms	79 sec
Better C Code	51 ms	51 sec

Table I Correlation Benchmarks
(C Compilers set to optimize for speed)

The image preprocessing and template extraction are performed by the 80486, the template correlation with the database is accelerated by using the VLSI image correlator, and postprocessing is subsequently performed by the 80486. The 80486 provides a flexible platform for general computation while the VLSI image correlator is fully optimized for a single operation, template correlation with the image database. The database correlation task is to compute the correlation of one template set against the entire database. The user's templates remain constant throughout the entire operation while the database templates varies as each known individual is considered in succession. Thus, the user's templates can be cached using local SRAM on the image coprocessor board to optimize the usage of the 8 MByte/sec ISA bus bandwidth (see Figure 7). Furthermore, since the image template data are only 8 bits wide, two templates can be transferred in parallel to take full advantage of the 16 bit data bus.

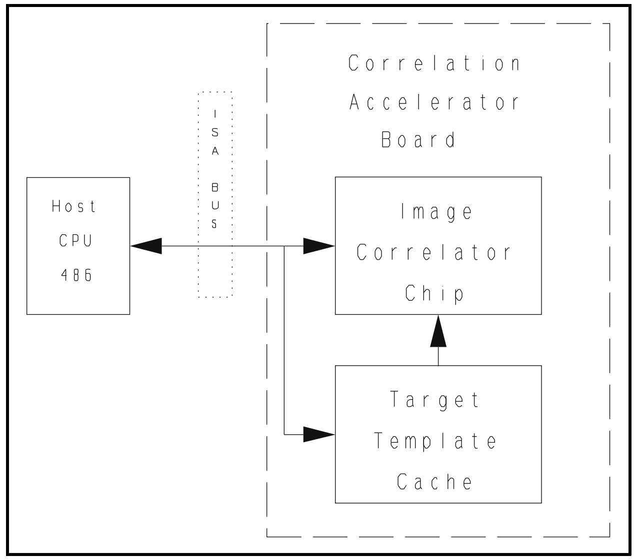

Figure 7 Correlation accelerator board - block diagram

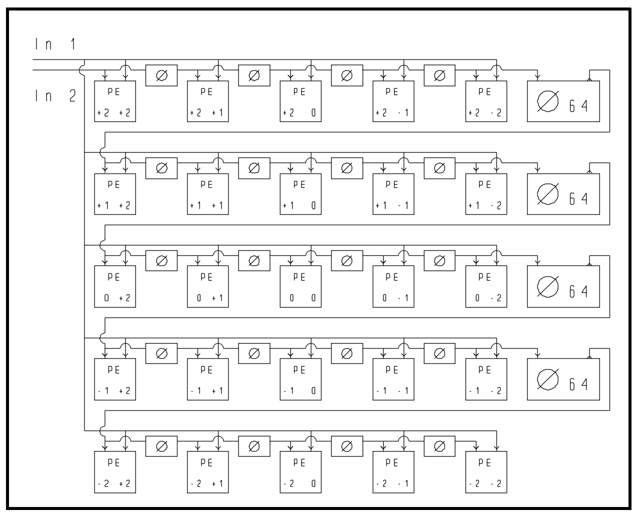

Figure 8 Pipelined Correlator Architecture
(Actual VLSI Chip Incorporated Two Correlators)

Thus, the VLSI correlator chip is designed with two independent image correlators such that two database entries can be correlated simultaneously over all 25 possible alignments. In this way, the correlation time per 4KByte template is reduced to 0.9 ms/template which increases the possible throughput of the VLSI image coprocessor system to about 1000 templates/sec. Thus, a moderately sized database of 500 persons (a few thousand images) can be completely correlated in a few seconds.

Image Correlator Architecture

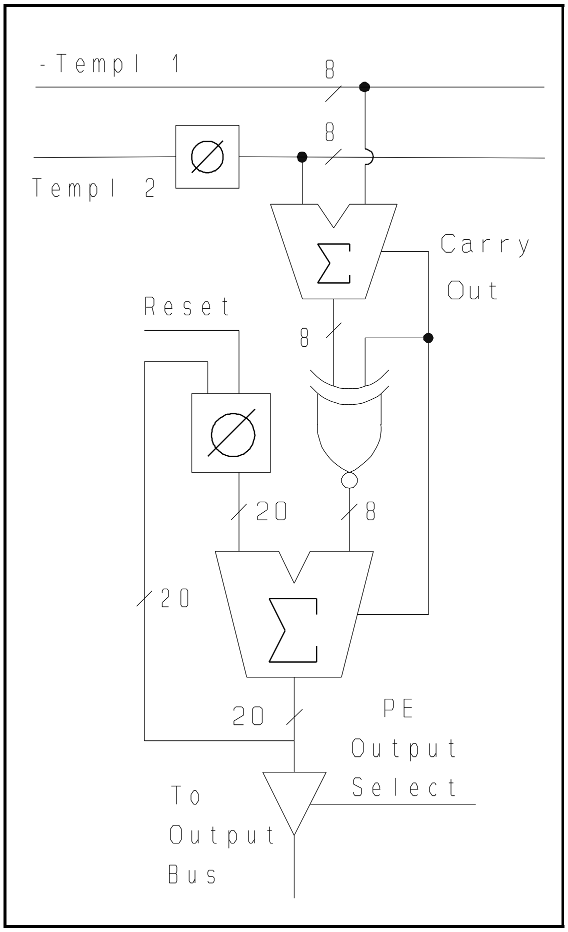

Figure 9 Processing Element

The architecture of the VLSI image correlator is optimized to perform absolute-difference correlations of a 68×68 template with 64×64 template over a range of alignments from +2 pixels to -2 pixels. The architecture is shown in Figure 8. A pixel value from template 1 is broadcast while pixels from template 2 are pipelined through the array of 25 processing elements (PE) during each clock cycle. Four 64 entry shift registers maintain proper pipelining of the 68×68 template. Each PE accumulates the absolute difference between its two input pixel values up to 20 bits at each clock cycle (see Figure 9). The size of the accumulator such as to guarantee that no overflow would occur during normal operation. After the two images have been processed through the correlator, each PE contains the absolute-difference correlation for its given offset (e.g. the upper left PE will contain the correlation of template 1 shifted by +2 pixels up and +2 pixels to the left.) The absolute-difference correlation scores are subsequently read out to the host processor. The minimum accumulated absolute-difference mismatch corresponds to the maximum correlation while the other correlation scores are ignored as being invalid due to template misalignment.

The actual VLSI chip contained two image correlators and was fabricated on a 6.8mm × 6.8mm die in a standard double metal, 2µm CMOS process through MOSIS (see Figure 10). The MAGIC layout editor was used to realize the fully custom design of the 60,000 transistor chip.

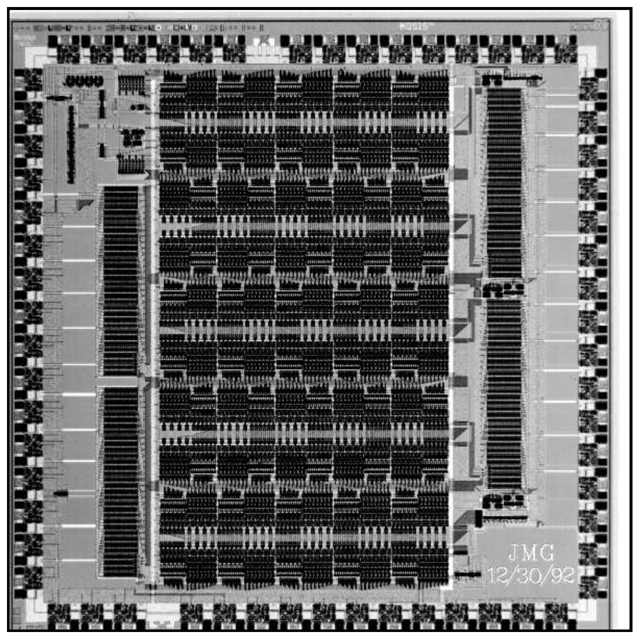

Figure 10 Die Photo

IV. System Performance

The real-time face recognition system user-interface is menu-driven and user-friendly. There are many additional features that were incorporated for rapid debugging, building of image databases, and development of more advanced recognition techniques. In all, the system software represents a large portion of the research effort and is implemented with approximately 40,000 lines of C and 80x86 assembly code. A typical screen capture of the real-time face recognition system is shown in Figure 11. The system initially locates the eyes of the user as shown by concentric circles overlaid on the original image. Subsequently, four small templates are extracted and compared to the database. The pseudo-scores of the top five candidates are shown at the bottom of the figure. The highlighted numbers indicate scores that exceed the threshold for a positive match. The darkened numbers indicate scores that exceed the threshold for a negative match. All match scores are normalized and offset such that the rejection threshold was 0 and the acceptance threshold was 100. Timing and memory requirements are shown in the text overlay below the extracted templates.

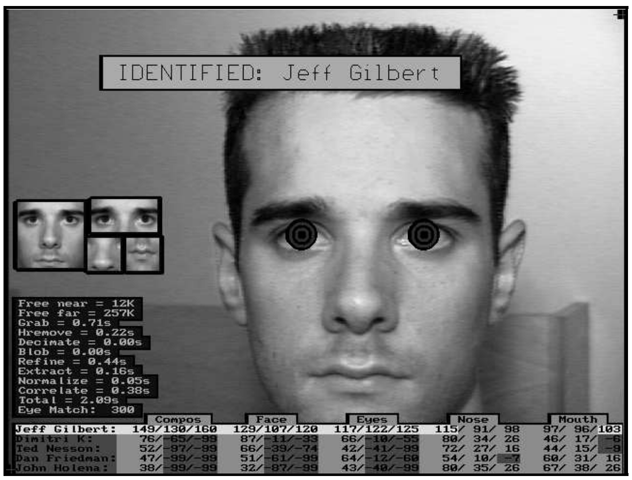

Figure 11 System in Action

The speed of the system is measured from when the image is presented to when the user is notified of an identification. During this time the system must digitize the video image through the frame grabber, locate the eyes, extract and normalize the templates, search the database via correlation, and interpret the correlation scores. The preprocessing and template extraction phase is performed using only the frame grabber and 80486/66DX2 in approximately 1.8 seconds and is independent of the database size. A typical timing breakdown for preprocessing and template extraction are shown in Table II. The template correlation is performed by the VLSI image correlator and depends on the size of the database. Typical database correlation time was approximately 0.3 seconds for a database of 173 images. Postprocessing is performed by the 80486 but is computationally quite simple and does not represent a significant portion of computing time.

The recognition performance of the system is highly dependent on the database of known persons and the

Grab Image from Frame Grabber	0.71 sec
Remove / Detect Highlights	0.44 sec
Decimate Image for Blob Location	0.02 sec
Blob Location	0.04 sec
Refinement	0.33 sec
Template Extraction	0.16 sec
Template Normalization	0.05 sec
Total	1.75 sec

Table II Typical Timings for Preprocessing

testing set. Cross-validation is a common technique for measuring recognition performance. The system was able to achieve a 88% recognition rate, a 93% correct matching with the top candidate, and a 97% correct matching with the top 3 candidates under cross-validation with a moderately varied database of 173 images of 34 persons.

However, cross-correlation provides a rather pessimistic estimate of recognition performance for this system as it does not take into account the fact that recognition occurs continuously every few seconds. Thus if the user is not recognized during one trial, he may tilt

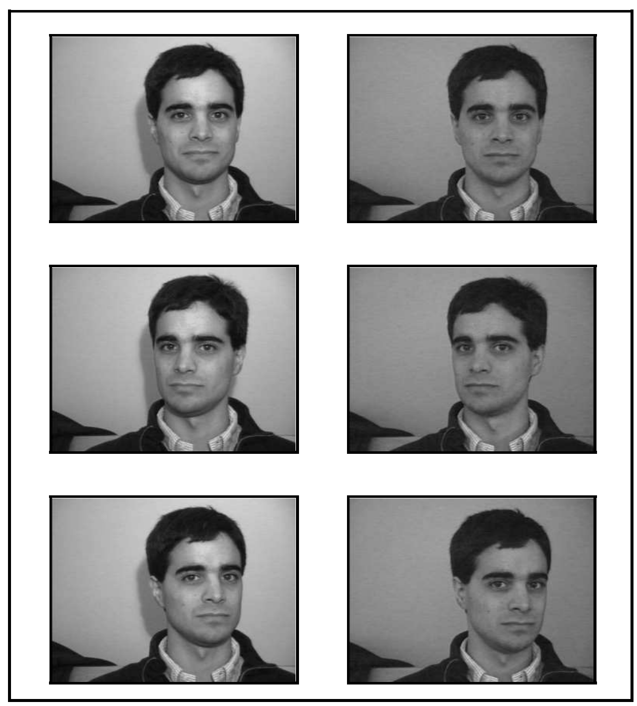
Figure 12 Typical Database Images

his head or move slightly so as to be recognized more readily on the next trial a few seconds later. Hence it is more important that the system does not mistakenly recognize a user as someone that they are not, than to miss the person and claim that they are not in the database. During actual usage, the system can sometimes require more than one trial, but recognition rarely takes more than three or four trials. Additionally, mistaken recognitions are also quite rare. As the recognition and rejection thresholds are adjustable, the trade-off between missing and mistakenly recognizing can be controlled to suit a particular application.

V. Conclusions

A real-time face recognition system was developed by making effective use of the computing power available from an IBM PC 80486 and by implementing a special purpose VLSI image correlator. The complete system requires 2 to 3 seconds to analyze and recognize a user after being presented with a reasonable frontal facial image. This level of performance was achieved through careful system design of both software and hardware. Issues ranging from algorithm development to software and hardware implementation, including custom digital VLSI design, were addressed in the design of this system. This approach of extremely focussed system software and hardware co-design can also be effectively applied to a wide range of high performance computing applications.

Acknowledgements

The authors would like to acknowledge the helpful discussions and contributions of Tomaso Poggio, Roberto Brunelli, Peter Hallinan, and Alice M. Chiang. In addition, support for VLSI chip fabrication was provided by the Massachusetts Microelectronics Center and by NSF through MOSIS. This work has been supported in part by an ARO Young Investigator Award for W. Yang, an NSF Young Investigator Award for W. Yang, a grant for High Performance Computing and Communications from NSF and ARPA, and a grant from the George W. Merck Fund at Community Funds.

References

[1] Robert J. Baron, "Mechanisms of human facial recognition," *International Journal of Man-Machine Studies*, vol. 15, pp. 137-178, 1981.

[2] Roberto Brunelli and Tomaso Poggio, "Face Recognition: Features versus Templates," Technical Report 9110-04, I.R.S.T, 1991.

[3] Peter J. Burt, "Smart Sensing within a Pyramid Vision Machine". *Proceedings of the IEEE*, 1988, vol 76, no 8, pp. 1006-1015.

[4] Jeffrey M. Gilbert, "A Real-Time Face Recognition System using Custom VLSI Hardware." Harvard Undergraduate Honors Thesis in Computer Science, 1993.

[5] Peter W. Hallinan, "Recognizing Human Eyes," SPIE Proceedings, vol. 1570, *Geometric Method in Computer Vision*, pp. 214-226, 1991.