Hardware Implementation for a Face Recognition Algorithm using Template Matching in Frequency Domain

PETER RAUSCHERT, MARCO KRIPS, ANTON KUMMERT AND YURI KLIMETS
Communication Theory, Department of Electrical and Information Engineering
University of Wuppertal
Rainer-Guenter-Strasse 21, 42119 Wuppertal
GERMANY

Abstract: - Based on video images, face identification gets more and more interesting to be implemented in new products. By the means of a very large scale integration (VLSI) based solution, this very challenging task can be solved. As target platform a field programmable gate array (FPGA) is used. Thus, the identification cycle time can be reduced by parallel processing. The realisable frame rate is very high and can be used easily for real-time applications. For identification purposes every picture is transformed using a fast fourier transformation (FFT). Working in fourier space offers 2 main advantages. By concentration on most significant frequency bands a considerable amount of memory can be saved and the system performance will be much better. Finally, identification is done by the use of compressed data and an euclidean classificator. All classification results are associated with confidence probabilities. Special parameters can be adapted in order to affect the required conditions.

Keywords: - Face Recognition, Template Matching, Frequency Domain

1 Introduction

Modern multimedia applications need new features to be more suitable for the customers. For nowadays applications, face identification, as a special part of image processing, gets more interesting. The range for applying such a feature is widely spread and can for example be found in automotive applications. Many different algorithms using biometric information have been developed. Most popular sources for biometric information are finger, iris, voice and face scans. Every source has its own salient features and can be chosen depending on boundary conditions. The latter are imposed by the essential requirements for an identification system. Those Systems requiring high security levels need a very expense identification process while systems with less strict restrictions can use simplified identification algorithms. In this work, image identification schemes are presented, offering good security features for applications without too critical requirements.

In the following, face identification is in the centre of our interest. Using this kind of biometric information leads to a system without need for a touch between the sensor and the users face. Thus, the users expense is minimized. Images can be taken by any kind of a camera source in different resolutions and colours. Due to our requirement for minimizing costs, our investigation is focused on greyscale data as being delivered by a charged coupled devices (CCD) camera.

The whole identification system persists of different stages for data processing. Two essential steps are the localisation and the identification of faces. First, a face graph must be located in a picture. After clipping, the identification algorithm is started. Although there are a lot of identification algorithms available, template matching applied to eyes or mouth is most common for localisation.

An important aspect for low cost applications are the hardware costs. Hence, hardware components have to be chosen, which can be manufactured in high quantity for low costs. Furthermore, hardware solutions offer considerable advantages with respect to processing speed due to parallel algorithms. To prove the usability for general hardware implementations our algorithm has been implemented in a field programmable gate array (FPGA).

2 Problem Formulation

The selection of a certain decision process has to be based on restrictions imposed by the final product.

2.1 Hardware Restrictions

The algorithm has been implemented by using logic devices. This target technology offers special characteristic properties and the used algorithm must be selected according to these requirements.

Thus, we are confronted with two fundamental restrictions concerning the choice of a suited algorithm :

- **Mathematical Functions :** The used mathematical operations have to be implemented in an efficient way. Therefore, the extensive use of exponential functions and floating point numbers will require expense use of hardware and should be avoided.
- **Data Reduction :** Caused by the huge amount of memory required for storing 2D data (here the data base), we need data compression. A suitable algorithm should focus on the relevant parts of the whole information. This allows use of smaller devices and saves time for identification.

Based on these conditions we evaluated a number of identification algorithms. The methods considered were template matching in fourier space [1], eigenfaces [2], geometrical features [3], and elastic bunch graph matching [4]. Due to the mentioned restrictions some algorithms, namely identification by geometrical features and elastic bunch graph matching, can be excluded from further treatment in our investigations.

2.2 Identification Behaviour

For additional analysis we investigated the statistical properties of the remaining algorithms.

One first aspect is the identification rate, which heavily depends on the data reduction factor. Reduction of data results in loose of information, so there is the need to make a compromise between this two facts.

Another important issue are the false acception rate (FAR) and false rejection rate (FRR). These error rates assess quality of the identification process. They are based on thresholds, while determine the tolerance of a system. The performance of a threshold can be investigated on basis of the claimed specifications.

Figure 1 shows behaviour of the error rates by changing the tolerance of a system. The FRR will drop down while the FAR rises by increasing the selected tolerance.

Since FAR and FRR are antidromic, the crossing point (equal error rate (EER)) is taken as quality measure.

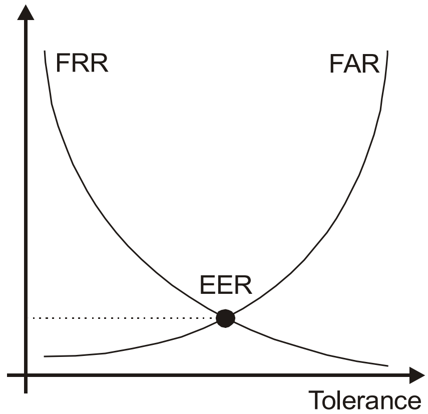

Fig. 1 : Error Rates

3 Problem Solution

Based on the claimed requirements, the fast fourier transformation (FFT) [1] using template matching was the best suited choice for such a problem. For explanation, the basic procedure is illustrated in the following.

An image of a face $\mathbf{I}(m,n)$, with $m \in [1,\ldots,M]$ and $n \in [1,\ldots,N]$, can be transformed into the frequency domain, using the discrete fourier transformation (DFT), by

$$\mathbf{F}(u,v) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\mathbf{I}(m,n)e^{-j2\pi\frac{mu}{M}}e^{-j2\pi\frac{nv}{N}} . \quad (1)$$

For our system, we used a padding algorithm to resize the image data to a fixed quadratic size, so that finally

$$M = N = 2^i \ , \ i \in \mathbb{N} . \quad (2)$$

After this adjustment, it is possible to use a fast fourier transformation (FFT) algorithm to realise DFT. By the use of a fast algorithm the number of multiplications can be reduced from $M \cdot N \cdot (M+N)$ to $M \cdot N \cdot ld(M \cdot N)$. By using FFT, computation complexity can be reduced considerably [6].

Obviously, spectra of different face images will be distinct. However, the variation among a large set of different face spectra will not be similar with respect to different spatial frequency values.

Usually, the variance of spectral values at low frequencies will be much more greater than that at high frequency values. Investigations by large sets of faces show an order by significance of different frequency values with respect to variance. Hence, to any frequency values of fourier space an order number can be assigned which is shown in figure 2. Obviously, low frequency values seems to be most significant, whereas high frequency values obtain a high ordering number.

With attention to the resulting symmetry in fourier space [5]

$$\mathbf{F}(u,v) = \mathbf{F}(u+M,v) = \mathbf{F}(u,v+N) \tag{3}$$

the assignment shown in figure 2 can be restricted to two quadrants.

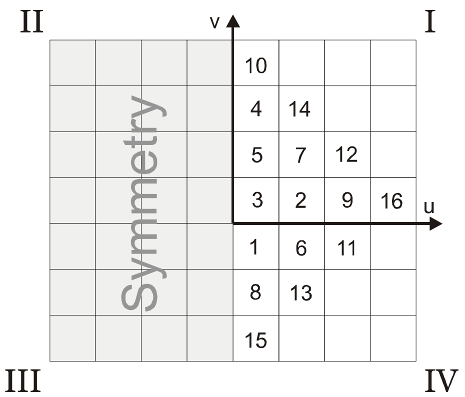

Fig. 2 : Ordering of Frequencies with Respect to Variance

As conclusion, the information degrades with rising spatial frequency values. The form can be described by a half rhomb, which opens around the constant component. Imaginary and real parts show similar behaviour and can be ordered similarly.

This number assignment procedure by information (variance) is a very important aspect. For identification, only a specific number K of frequencies are used for classification in order to reduce memory and save processing time. From the 2D array of spectral values $\mathbf{F}(u,v)$ we generate a $K \times 1$ vector $\mathbf{F}(k)$, $k = [1,\ldots,K]$, where the K relevant frequencies are stored in the order of their associated ordering number

$$\mathbf{F}(u,v) \Rightarrow \mathbf{F}(k)\,. \tag{4}$$

Let us summarize, the padded image $\mathbf{I}(m,n)$ can be transformed into frequency domain by using FFT. Afterwards, the investigations are concentrated onto K coefficients. Also the data base $\mathbf{DB}_j(k)$, containing J known faces, is compressed to K coefficients per person and will be saved in a matrix.

For the final classification (main identification) it is necessary to sum the square of difference between $\mathbf{DB}_j(k)$ and $\mathbf{F}(k)$.

$$\delta_j = \sum_{k=1}^{K} (\mathbf{F}(k) - \mathbf{DB}_j(k))^2 \ \ , \ j \in [1,\cdots,J]\,. \tag{5}$$

Finally, the minimum distance $\delta_{\min}$ indicates the person of the data base which has been identified. Furthermore, optional thresholding can be added to influence the error rates.

4 FPGA Implementation

4.1 Basics

First of all, a software based modelling of the algorithm has been performed by using Matlab®.

Basic characteristics of the identification process can be investigated by using this model. These are necessary to configure the hardware based model according to our requirements.

First, the accurate ordering (see figure 2) for the coefficients was specified. Second, the influence of the image size to quality of face identification is another important issue. Our investigations showed, that a picture size of 32×32 pixel is a suitable compromise between need of memory - associated with processing time - and loss of information.

4.2 Hardware Description

The whole design was done by using VHDL (very high speed integrated circuits (VHSIC) hardware description language) [7]. This is a standard description language (IEEE 1076), which is not restricted in terms of target technology. Thus, it offers a great flexibility.

In our case, an FPGA solution has been selected. The advantage for basic investigations can be seen in the fact that FPGAs are reconfigurable and can be updated easily with new versions of the face identification algorithm.

Identification can be subdivided into two significant components, namely transformation and classification (see figure 3). First, the data is transformed and second the classification is done by choosing some of the spatial frequencies.

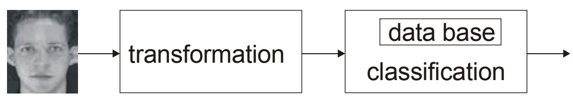

Fig. 3 : Identification Flow

The setup for our design is done similar and will be explained according to this block representation.

4.3 Transformation

2D transformation can be done by a bunch of 1D transformations. In our case, we need 64 single 1D FFTs to get the spatial frequencies from a facial image scan of size 32×32 pixel. The transformation is done row-by-row and column-by-column as shown in Figure 4.

The 'Transposed Memory' and 'Spatial Frequencies' named random access memories (RAMs) must be designed for storing complex data. Thus, two RAMs, one for real and one for imaginary part, are connected to one complex RAM element. Several RAMs will have access to two common buses, thus the RAMs need a tristate driven output.

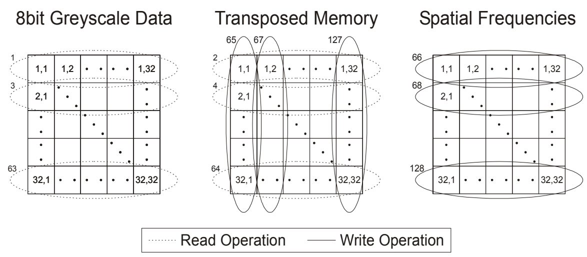

Fig. 4 : Memory Configuration

Block RAM is provided by some FPGAs e.g. Xilinx® Virtex™ and Virtex-E™ series and can be used to save logic devices.

For implementing a 1D FFT transformation, the Xilinx® core generator was used. The component ports are handling integer numbers for input and output data. All calculations for realising exponential functions are done internally by using fixed point numbers with a user-defined precision.

Finally, there is a need for a controlling device, which monitors data processing. For this task was designed a finite state machine (FSM) to control the whole flow of transformation.

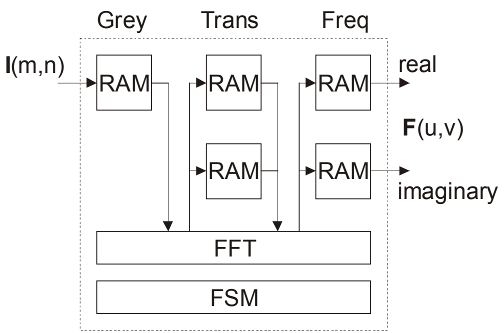

Fig. 5 : FFT Configuration

Figure 5 summarizes the configuration using all components (RAMs, FFT, FSM) in a simplified block diagram. Real and imaginary parts are treated separately.

4.4 Classificator

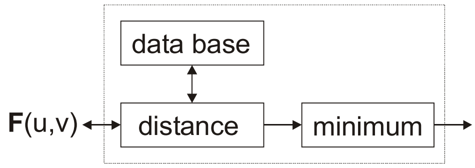

Fig. 6 : Classificator Configuration

In our system, the classificator consists of 3 main components (see figure 6) :

1. All compressed reference data is stored in the 'data base'. In our case, data base is of size 20×50 elements. Every set of data for one image consists of 20 coefficients. Such a compression is done for 50 different reference images. If there are 10 known users every person is represented by 5 samples of data.
2. For classification, data is read from the previous performed transformation block and compared to the data base content. The coefficients for $\mathbf{F}(u,v)$ are chosen according to the ordering, which was saved into a table inside the 'distance' component.
3. Finally, the 'minimum' distance is found. Thresholding the minimum distance proves the result as a valid classification.

5 Results

As discussed above, a face identification algorithm is implementable in hardware. The most suitable solution is the template matching in frequency domain. It offers a high data reduction factor. Without scaling, there is a need for word length of about 20 bit for frequency data. The greyscale data is using words with 8 bit length. Thus, the data compression factor is given by

$$\frac{32 \cdot 32 \cdot 8bit}{50 \cdot 20bit} = 20.48 \quad . \tag{6}$$

Memory size of the data base is highly reduced using this kind of data compression.

For our design, we investigated the difference between the mathematical model using Matlab® and the VHDL hardware based solution. To rate the quality of transformation, we compared the established minimal difference $\delta_{\min}$ in both models. As shown in Table 1 , both solutions show a very similar behaviour. In 80% of all cases the difference is just 4% or less.

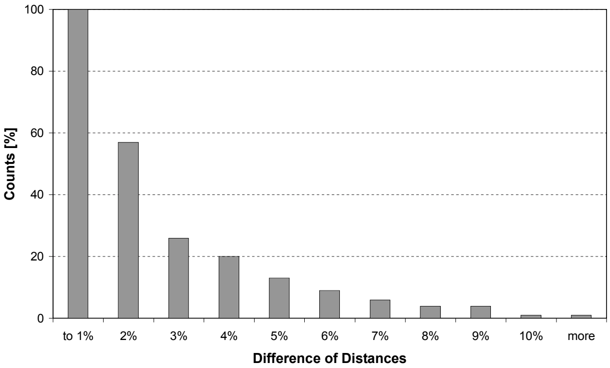

Table 1 : Differences between Models

This constitutes that the results for the identification rate are very similar. In figure 7 identification rates for both models, depending on data compression factor, are shown. The number of relevant coefficients is taken as measure for data compression.

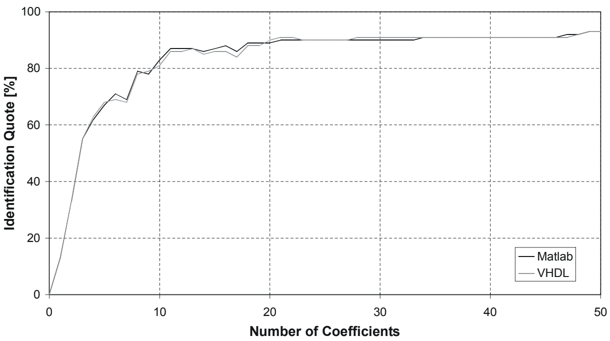

Fig. 7 : Identification Rates

Near the origin, the graph in figure 7 rises quickly. But it can be seen easily, that the identification quality will not be improved highly by using more than 20 coefficients. Obviously, 20 relevant coefficients turned out to be sufficient.

The identification rate is around 90% in this configuration. Thus, 9 identifications out of 10 are done correctly in average. For 'equal error rate' a result around 13% can be asserted. As conclusion, this design is suited for applications with moderate security requirements only.

The synthesis for a Xilinx® FPGA XCV1000 showed a utilization level of 37,84% for configurable logic devices (CLB) and 96,875% for block RAMs.

A critical path analysis showed a processing time t_{proc} of 43.36ns. Following this result, the system clock can be set to 23,06 MHz for the design. A complete identification takes 14207 clocks. Using a 23 MHz clock frequency, 1618 identifications per second can be done. Looking on relevant applications, this processing speed is more than sufficient. The operating efficiency of this algorithm, if implemented on an FPGA, is obvious.

6 Conclusion

In this paper, a pretentious hardware implementation for a face identification problem is discussed. An appropriate solution is based on a template matching algorithm.

FPGAs offer the ability of a great speed-up by using parallel data processing. Furthermore, the configuration of the FPGA can be fit to special specifications just by updating the setup.

First results obtained by an FPGA-based implementation of a face identification algorithm have been presented. Nevertheless, further investigations can be useful for improving results. The next step will be to realise an artificial neural network (ANN) to improve the classificator based on the same given information.

References :

[1] H. Spies, I. Ricketts, Face Recognition in Fourier Space, *Proc. Vision Interface'2000, Montreal*, 2000, pp. 38-44.

[2] M. Turk, A. Pent land., Face Recognition using Eigenfaces, *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586-591.

[3] R. Brunelli, T. Poggio, Face Recognition through Geometrical Features. *Proceedings of ECCV '92*, 1992, pp. 792-800.

[4] L. Wiskott, J.M. Fellous, N. Krüger, C. von der Malsburg, Face Recognition by Elastic Bunch Graph Matching, *CAIP'97*, 1999.

[5] W.K. Pratt, D*igital Image Processing,* John Wiley and Sons Inc., 1978.

[6] J. W. Cooley, J. W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series, *Mathematics of Computation*, Vol.10, 1965, pp. 297-301.

[7] P.J. Ashenden, *The Designer's Guide to VHDL*, Morgan Kaufman Publishers, 1999.