A SMART CAMERA FOR FACE RECOGNITION

Martijn Reuvers[1], Richard Kleihorst[2], Harry Broers[3], Ben Kröse[1]

[1]University of Amsterdam, NL
[2]Philips Research Laboratories, Eindhoven,NL
[3]Philips CFT, Eindhoven, NL
E-mail: mreuvers@science.uva.nl

ABSTRACT

There is a rapidly growing demand for using smart cameras for various biometric applications in surveillance. Although having a small form-factor, most of these applications demand huge processing performance for real-time processing. Face recognition is one of those applications. In this paper we show that we can run face recognition in real-time by implementing the algorithm on an architecture which combines a parallel pixel processor with a digital signal processor. Everything fits within a digital camera, the size of a normal surveillance camera.

I. INTRODUCTION

Recently, face recognition is becoming an important application for smart cameras. Face detection and recognition requires lots of processing performance if real-time constraints are taken into account [1].

We want to show in this publication that it is possible using thought-over smart camera architectures to achieve good, real-time face recognition results. A "smart camera" is hereby defined as a stand-alone device which is preferably programmable with a size not bigger than a typical video surveillance camera.

The platform we suggest for face recognition is the Intelligent Camera (INCA$^+$) produced by Philips CFT [2] as shown in Figure 1. This camera houses a CMOS sensor, a parallel processor for pixel crunching and a DSP for the high level programs. We will show in this paper that this platform is ideal for face recognition.

The contents of the paper is as follows: In Section II we explain about the architecture of the camera, In Sections III and IV respectively, we explain about the algorithms that we used for recognition. The results are given in Section V and conclusions are drawn in Section VI.

Fig. 1. INCA$^+$ camera

II. MOTIVATION OF THE ARCHITECTURE

"Face recognition" consists of a face *detection* and a face *recognition* part. In the detection part faces are detected in the scene and their Region of Interest (ROI) are forwarded to the face recognition process where the found faces are matched to a database in order to recognize and identify them.

The detection part is face oriented (high-level image processing). It finds face candidates in the scene. In order to reduce the amount of work, the image needs to be preprocessed by a number of low-level operations. These low-level operations are at pixel level, simple and equal for each pixel. This allows massive data-level parallelism. So the detection part involves low-level as well as high-level image processing.

The recognition part uses high-level image processing and only works on a few faces per second, but it has a high amount of operations in an iterative way while a database is "scanned". Because of the higher complexity of the instructions and the combination with an operating system, this part of the algorithm is best mapped on a task-parallel architecture.

The different aspects of the two algorithmic tasks (low and high-level image processing) have made us choose for a dual processor approach. The low-level image preprocessing approach of the face detection part is mapped on a massively parallel processor “Xetal” [3] working in SIMD (Single Instruction Multiple Data) mode. The high level image processing approach of the detection and recognition part is mapped on a high-performance fully programmable DSP core “TriMedia” [4]. This DSP has a VLIW (Very Long Instruction Word) architecture where instruction fetch, data fetch and processing are performed in a pipelined fashion.

For the defined task, the two processors can be simply connected in series (see Figure 2). The Xetal does face detection preprocessing, the TriMedia does the actual face detection and recognition. The operating system also runs on the TriMedia.

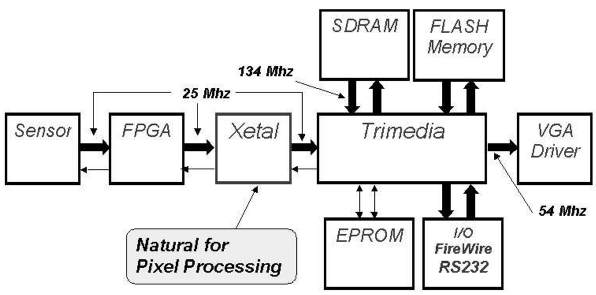

Fig. 2. INCA^{+} Architecture

III. FACE DETECTION

In the face detection part we take an image from the sensor and detect and localize an unknown number (if any) of faces.

Popular techniques to detect faces in an image often involve skin-color segmentation [5], [6]. This segmentation is based on the fact that the color of the human skin resides in a well defined area of a chosen color model. Skin-color segmentation is fast but can only be done in well defined surroundings. If the light color changes in a certain surrounding the technique will fail since the well defined skin-color area also changes.

We want our detection algorithm to be robust even under totally different lighting conditions, so we implemented the Haar-Face detection algorithm as described in [7] and [8]. This image-based detection algorithm has proven itself robust under various lighting conditions. Since it is image-based, thus working at picture structure level instead of at pixel level, even hand drawn images of faces can be detected (see Figure 6).

Before the image is transferred to the TriMedia the image is preprocessed by the Xetal processor. First Xetal converts the RGB color-image to a gray-level image. After that Xetal performs lighting correction to improve the quality of the image and thus the quality of detection. Xetal also performs Canny edge preprocessing to reduce the number of face candidates, and Xetal calculates two so-called integral images as described in [7] and [9]. One for the lighting corrected image and one for the Canny edge image. The TriMedia receives the original gray-level image as well as the two integral images from the 3 communication channels. So the TriMedia saves valuable processor time since it won’t have to calculate all these images itself.

Fig. 3. Input scene. Note that the small image on the phone display is a face.

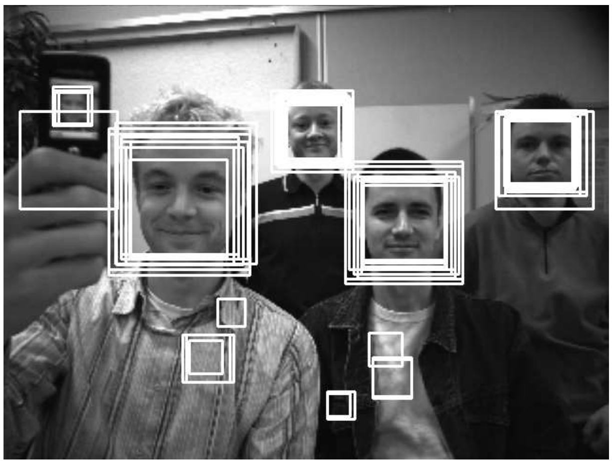

Fig. 4. Haar-Face detection; No grouping.

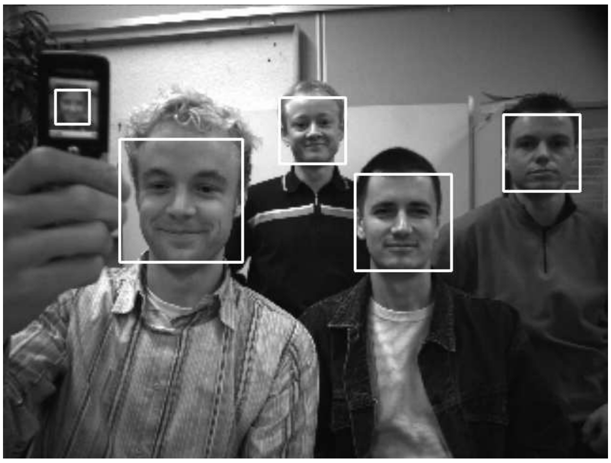

Fig. 5. Haar-Face detection; Grouping. Note that the small face on the phone is also recognized correctly.

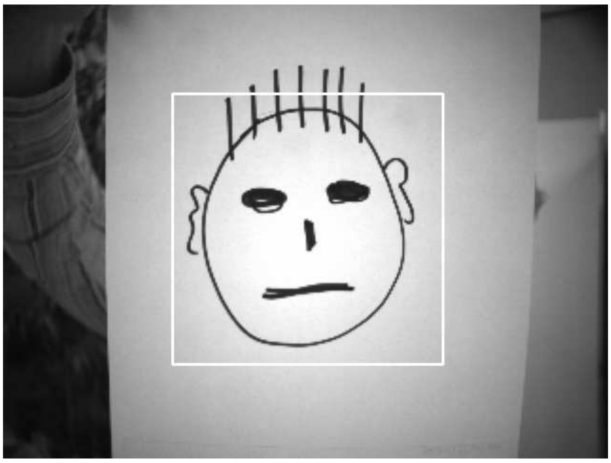

Fig. 6. The Haar-Face detection method is image-based, so even more abstract concepts are recognized.

The detection algorithm runs on TriMedia and uses the lighting corrected integral image. It scans the image at several different hierarchy levels and returns every possible face candidate as shown in Figure 4. It uses the Canny integral image to speed up the detection process even further [9].

After we obtained all possible face candidates we apply a grouping algorithm to reduce a group of face candidates into one positive detection (see Figure 5).

Horizontally and vertically through the face image, a gray-level projection is performed whose minima enable the detection of the position of the eyes in order to normalize the face image around the eye positions before feeding it to the recognition phase [10], [11].

The normalized images, which only contain the eyes and part of the nose, will be 150×60 pixels. These images are forwarded to the recognition phase.

IV. FACE RECOGNITION

For face recognition a Radial Basis Function (RBF) neural network is used [12]. The reason behind using an RBF neural network is its ability for clustering similar images before classifying them. RBF based clustering received wide attention in the neural networks community. Apart from good clustering capabilities RBF networks have a fast learning speed, and a very compact topology.

An RBF neural network structure is demonstrated in Figure 7. Its architecture is similar to that of a traditional three-layer feed forward neural network. The input layer of this network is a set of n units, which accepts the pixels of the face ROI which is gained from the face detection part. Since it is normalized with a $150 * 60$ pixel face, it follows that $n = 9000$.

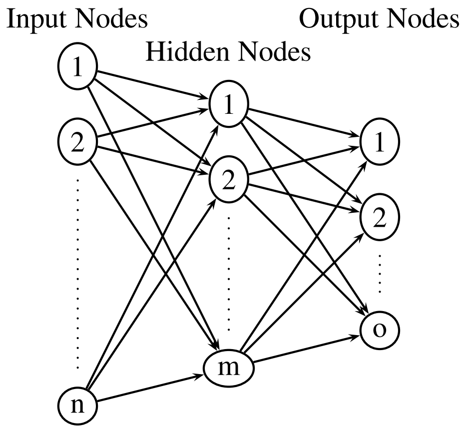

Fig. 7. *Architecture of an RBF Neural Network*

The input units are completely connected to the hidden layer with m hidden nodes. Connections between the input and the hidden layers have only offsets. The purpose of the hidden layer is to cluster the data and decrease its dimensionality. The RBF hidden nodes are also completely connected to the output layer.

The number of outputs depends on the number of people to be recognized (for example, for 100 persons $o = 100$). The output layer provides the response to the activation pattern applied to the input layer.

The activation function (basis function) of the hidden units is known by the distance between the input vector and a prototype vector. It is stated as follows

[13]:

$$F_i(x) = G_i(\| x - c_i \|^2 / \sigma_i), \quad i = 1, 2, \ldots, m \tag{1}$$

where x is an n-dimensional input feature vector, c_i is an n-dimensional vector called the center of the RBF hidden node, σ_i is the width (also called radius) of the node and m is the number of the hidden nodes. The activation function G of the hidden nodes is a Gaussian with mean vector c_i and variance σ_i as follows:

$$F_i(x) = e^{\left(-\frac{\|x-c_i\|^2}{\sigma_i^2}\right)}, \; i = 1, .., m \tag{2}$$

The response of the k'th output unit (among the o number of outputs) for input x is given as:

$$Out_k(x) = B_k + \sum_{i=1}^{m} F_i(x) * W(i,k), \; k = 1, .., o \tag{3}$$

where $W(i,k)$ is the connection weight of the i'th RBF hidden node to the k'th output node and B_k is the bias of the k'th output. A value close to 1 indicates a positive identification.

V. OVERALL PRACTICAL PERFORMANCE

Our algorithms have been mapped to a handheld camera device as shown in Figure 1. The face detector detects faces at a rate of 4 frames/sec. There can be any number of faces on each frame. Combined with the recognition part the system runs at approximately 2 frames/sec.

The operating system obtains the IDs of the recognized person and monitors the reliability of recognition as reported by the face recognition part. If this is high enough, a person is positively identified and will also not be reported in subsequent frames until he/she leaves the scene or another person shows up.

VI. CONCLUSIONS AND FUTURE WORK

Face recognition is becoming an important application for smart cameras. However, up till now, the processing required for real-time detection, prohibited integration of the whole application into a small sized, consumer type of camera. This paper showed that by:

1. Proper selection of algorithms, both for face detection and recognition,
2. Adequate choice of processing architecture, supporting both SIMD and ILP types of parallelism,
3. Tuning the mapping of algorithms to the selected architecture,

this integration can be achieved. We implemented the algorithms on a small smart camera. As a result we can recognize one face per 500 ms.

Future research will focus on further tuning the mapping of the algorithms and achieving speedups for large databases.

VII. REFERENCES

[1] B. L. E. Hjelmas, "Face detection: a survey," *Computer Vision and Image Understanding*, vol. 83, pp. 236–274, 2001.

[2] Centre For Industrial Technology. http://www.cft.philips.com/, 2003.

[3] A. Abbo and R. Kleihorst, "Smart cameras: Architectural challenges," in *Proceedings of ACIVS 2002 (Advanced Concepts for Intelligent Vision Systems)*, (Gent, Belgium), 2002.

[4] TriMedia Technologies. http://www.trimedia.com, 2003.

[5] T. Majoor, "Face detection using color based region of interest selection," tech. rep., University of Amsterdam, Amsterdam, NL, 2000.

[6] R.L.Hsu, M.Abdel-Mottaleb and A.K.Jain, "Face detection in color images." http://www.cse.msu.edu/˜hsureinl/facloc/index_facloc.html, 2003.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[8] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *IEEE ICIP*, vol. 1, pp. 900–903.

[9] Intel Open Computer Vision Library. http://sourceforge.net/projects/opencvlibrary, 2003.

[10] F. Zuo and P. H. de With, "Fast human face detection using successive face detectors with incremental detection capability," *Proc. SPIE*, no. 5022, 2003.

[11] R. Kleihorst *et al.*, "An SIMD smart camera architecture for real-time face recognition," in *Abstracts of the SAFE & ProRISC/IEEE Workshops on Semiconductors, Circuits and Systems and Signal Processing*, (Veldhoven, The Netherlands), Nov 26–27, 2003.

[12] J. Haddadnia, K. Faez, and P. Moallem, "Human face recognition with moment invariants based on shape information," in *Proceedings of the International Conference on Information Systems, Analysis and Synthesis*, vol. 20, (Orlando, Florida USA), International Institute of Informatics and Systemics (ISAS'2001), 2001.

[13] Y.-H. Hu and J.-N. Hwang, eds., *Handbook of neural network signal processing*. CRC Press, 2002.