8

Design, Implementation and Evaluation of Hardware Vision Systems dedicated to Real-Time Face Recognition

Ginhac Dominique, Yang Fan and Paindavoine Michel
LE2I - University of Burgundy
France

1. Introduction

Human face recognition is an active area of research spanning several disciplines such as image processing, pattern recognition, and computer vision. Different techniques can be used to track and process faces (Yang et al, 2001), e.g., neural networks approaches (Férand et al., 2001, Rowley et al., 1998), eigenfaces (Turk & Pentland, 1991), and the Markov chain (Slimane et al., 1999). Most researches have concentrated on the algorithms of segmentation, feature extraction, and recognition of human faces, which are generally realized by software implementation on standard computers. However, many applications of human face recognition such as human-computer interfaces, model-based video coding, and security control (Kobayashi, 2001, Yeh & Lee, 1999) need to be high-speed and real-time, for example, passing through customs quickly while ensuring security.

Liu (1998) realized an automatic human face recognition system using the optical correlation technique after necessary preprocessing steps. Buhmann et al. (1994) corrected changes in lighting conditions with an analog VLSI silicon retina in order to increase the face recognition rate. Matsumoto & Zelinsky (2000) implemented in real time a head pose and gaze direction measurement system on the vision processing board Hitachi IP5000.

For the last years, our laboratory has focused on face processing and obtained interesting results concerning face tracking and recognition by implementing original dedicated hardware systems. Our aim is to implement on embedded systems efficient models of unconstrained face tracking and identity verification in arbitrary scenes. The main goal of these various systems is to provide efficient robustness algorithms that only require moderated computation in order 1) to obtain high success rates of face tracking and identity verification and 2) to cope with the drastic real-time constraints.

The goal of this chapter is to describe three different hardware platforms dedicated to face recognition. Each of them has been designed, implemented and evaluated in our laboratory. In a first part, we describe a real time vision system that allows the localization of faces and the verification of their identity. This embedded system is based on image processing techniques and the radial basis function (RBF) neural network approach. The robustness of this system has been evaluated quantitatively on real video sequences. We also describe three hardware implementations of our model on embedded systems based, respectively, on field programmable gate array (FPGA), zero instruction set computer (ZISC) chips, and

Source: Face Recognition, Book edited by: Kresimir Delac and Mislav Grgic, ISBN 978-3-902613-03-5, pp.558, I-Tech, Vienna, Austria, June 2007

digital signal processor (DSP) TMS320C62.We analyze the algorithm complexity and present results of hardware implementations in terms of resources used and processing speed.
In a second part, we describe the main principles of a full-custom vision system designed in a classical 0.6 μm CMOS Process. The development of this specific vision chip is motivated by the fact that preliminary works have shown that simplified RBF networks gave interesting results but imposed a fast feature extraction to reduce the size of the input vectors of the RBF network. So, in order to unload a consequent calculation part of FPGA, we have decided to design an artificial retina embedding the extraction of input vectors of RBF network. For this purpose, a VLSI sensor is proposed to realize the image acquisition, to extract a window of interest in the whole image, to evaluate the RBF vectors as means values of consecutive pixels on lines and columns. A prototype based on this principle, has been designed, simulated and evaluated.
In a third part, we describe a new promising approach based on a simple and efficient hardware platform that performs mosaicking of panoramic faces. Our objective is to study the panoramic face construction in real time. So, we built an original acquisition system composed of five standard cameras, which can take simultaneously five views of a face at different angles. Then, we chose an easily hardware-achievable algorithm, based on successive linear transformations, in order to compose a panoramic face from the five views. The method has been tested on a large number of faces. In order to validate our system, we also conducted a preliminary study on panoramic face recognition, based on the principal-component method. Experimental results show the feasibility and viability of our system.
This rest of the chapter is organized as follows. Section II, III and IV describe the three systems designed by our team. In each of these sections, we present the principles of the system, the description of the hardware platform and the main simulated and experimental results. Finally, the last section presents conclusion and future works.

2. Real-time face tracking based on a RBF Neural Network

Face recognition is a very challenging research problem due to variations in illumination, facial expression and pose. It has received extensive attention during the past 20 years, not only because of the potential applications in fields such as Human Computer Interaction, biometrics and security, but also because it is a typical pattern recognition problem whose solution would help in solving other classification problems.
The recognition technique used in this first embedded system is based on Radial Basis Function (RBF) networks. The RBF neural networks have been successfully applied to face recognition. Rosenblum et al. (1996) developed a system of human expressions recognition from motion based on RBF neural network architecture. Koh et al. (2002) performed an integrated automatic face detection and recognition system using the RBF networks approach. Howell & Buxton (1998) compared RBF networks with other neural network techniques on a face recognition task for applications involving identification of individuals using low-resolution video information. The RBF networks give performance errors of only 5%–9% on generalization under changes of orientation, scale, pose. Their main advantages are computational simplicity and robust generalization. Howell and Buxton showed that the RBF network provides a solution which can process test images in interframe periods on a low-cost processor. The simplicity and the robust generalization of the RBF networks approach, with its advantages due to the fact that it can be mapped directly into the existing neural networks chips lead us to elaborate our model using a RBF classifier.

We chose three commercial embedded systems for hardware implementations of face tracking and identity verification. These systems are based, respectively, on most common electronic devices: FPGA, zero instruction set computer (ZISC) chips, and digital signal processor (DSP) TMS320C62. We obtained processing speeds of, respectively, for three implementations: 14 images/s, 25 images/s, and 4.8 images/s.

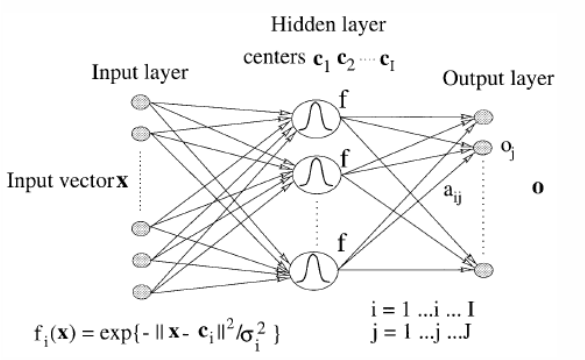

Figure 1. Radial basis function neural network

2.1 Description of the RBF model

The RBF neural network (Park & Sandberg, 1991) has a feedforward architecture with an input layer, a hidden layer, and an output layer as shown in Figure 1. The input layer of this network has *N* units for an *N*-dimensional input vector. The input units are fully connected to the hidden layer units, which are in turn connected to the *J* output layer units, where *J* is the number of output classes. RBF networks belong to the category of kernel networks. Each hidden node computes a kernel function on input data, and the output layer achieves a weighted summation of the kernel functions. Each node is characterized by two important associated parameters: 1), its center and 2) the width of the radial function. A hidden node provides the highest output value when the input vector is close to its center and this output value decreases as the distance from the center increases. Several distances can be used to estimate the distance from a center but the most common is the Euclidean distance d(x). The activation function of the hidden node is often a Gaussian function such that each hidden node is defined by two parameters: its center c_i and the width of the radial function σ_i.

$$d(x) = \|x - c_i\| \qquad f_i(x) = \exp^{(-d(x)^2/\sigma_i^2)} \tag{1}$$

The training procedure undergoes a two-step decomposition: estimating c_i and σ_i and estimating the weights between the hidden layer and output layer. The estimation of these parameters is largely detailed in Yang & Paindavoine (2003).

2.2 Description and test of our model

Many face recognition algorithms require segmenting the face from the background, and subsequently extracting features such as eyes, nose, and mouth for further processing. We propose an integrated automatic face localization and identification model only using a classifier which responds to the question, "Does the input vector correspond or not the

person to be verified?" The idea behind this is to simplify the model and reduce computation complexity in order to facilitate hardware implementations.

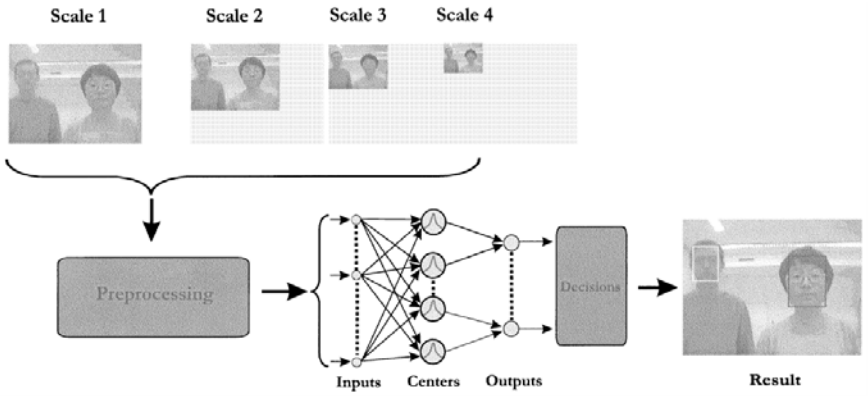

Figure 2. Structure of the face tracking and identity verification model

Figure 2 represents the structure of our model. The size of faces in the scene varies from 40 x 32 pixels to 135 x 108 pixels with four scales. The ratio between any two scales is fixed to 1.5 (Howel & Buxton, 1998).We first subsample the original scene and extract only the 40 x 32 windows in the 4 subsampled images. Each pre-processed 40 x 32 window is then fed to RBF network as an input vector. After the training procedure, the hidden nodes obtained are partially connected to the output layer. In fact, the hidden nodes associated with one person are only connected to the output node representing this class. This technique reduces data dependencies and is computationally more efficient (Koh et al., 2002). The decision stage yields the presence, position, identity and scale of the faces using the maximal output values of the RBF neural network.

In order to evaluate and validate our model, we made experiments based on video sequences of 256 images. In all sequences, the scene size is 288 x 352 pixels and they are zero, one, two, or three different faces presented (see. Figure 4). We have decided to verify two persons in these sequences. The 12 same training faces (see Figure 3) are used in order to compare the different configurations of the model.

Figure 3. 2x12 learning faces

First, in order to simplify future hardware implementations, the first phase has consisted in reducing the input vectors length of the RBF network. In the preprocessing stage, we use first all pixels of each 40 x 32 window to compose the feature vectors. Each pixel represents one component of the vector. So, the input vectors of RBF neural network have 40 x 32 components. Second, we minimize the number of components in order to reduce the

computing time. We realize a subsampling preprocessing: sample one pixel out of 4, 8, and 16 on each row of each window. We display some tested images (see Figure 4).

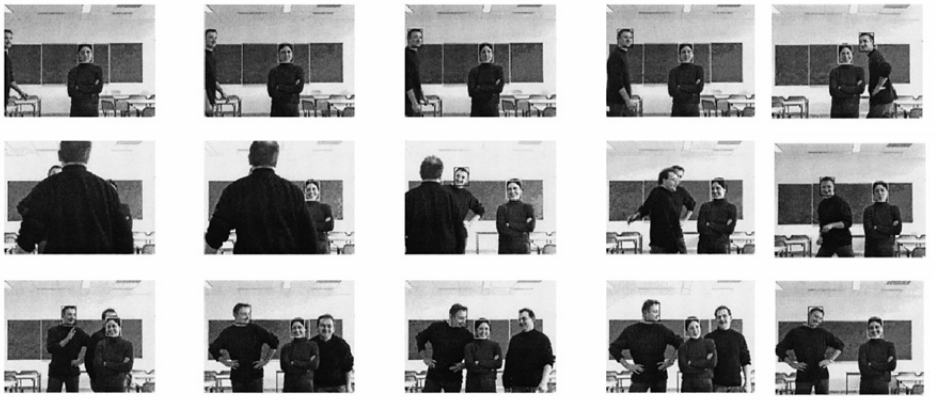

Figure 4. Some results of face tracking and identity verification

Results of face tracking and identity verification reveal that performances decreases quickly when the input vectors have 80 components. In fact, incorrect detection regularly appears when we use only one pixel out of 16 on each row of a window. The best results are obtained with one pixel out of four using the Euclidean distance $d_2(x)$ to compute the difference between an input vector and the centers (kernels) for each hidden node of the RBF neural network (see Eq. 2). The distance $d_1(x)$ is usually better when we use some noisy images (Sim et al., 2000). Another distance considers only the components whose difference between x_n and c_n is greater than a threshold δ. Here, the threshold δ has been regulated to 10. The experiments show that we have the best result with the $d_0(x)$ distance.

$$d_2(x) = \sqrt{\sum_{1 \le n \le N} (x_n - c_n)^2} \quad d_1(x) = \sum_{1 \le n \le N} |x_n - c_n| \quad d_0(x) = \sum_{1 \le n \le N} 1 \ \forall |x_n - c_n| > \delta \tag{2}$$

Finally, we have evaluated some variations of the RBF kernel activation functions. The Gaussian function is usually taken as the kernel activation function (see. Eq. 1) where d(x) is the measured distance between the input vector x and the center c. Another approach is the use of a simplified activation, for example the replacement of the Gaussian function in the RBF network by a Heaviside function leading to a simplified hardware implementation. The width of this function is the width σ associated to the corresponding center.

$$f(x) = \begin{cases} 1 & d(x) \le \sigma \\ 0 & d(x) > \sigma \end{cases} \tag{3}$$

The number of no-detections has increased with the Heaviside function. The rate of correct results decreases from 98.2% to 93.4%. In fact, the RBF neural network using the Heaviside function restrains the capacity of generalization by lack of interactions between centers of a same class: the model only detects faces that are sufficiently close to training examples.
Among all configurations of the model, the best performance has been obtained with 320 components of input vectors (subsampling 1 pixel/4 on each row of a window), using

measured distance $d_0(x)$ and the Gaussian activation function: the success rate is 98.2%. Almost all the faces are well detected, localized, and identified in sequences of images.

2.3 Hardware Implementations

Hardware implementations of the RBF approach have been realized for different applications, on either FPGA (Pérez-Uribe & Sanchez, 1996), or neurochip (Skrbek, 1999). Commercial RBF products include the IBM ZISC chip and the Nestor Ni 1000 chip (Lindbalad et al., 1995). Here, our aim is to elaborate in real time an efficient model of unconstrained face tracking and identity verification in arbitrary scenes. Thus, hardware implementations have been realized on three embedded systems based on FPGA, ZISC chip, and DSP. We use industrial electronic systems: a MEMEC board, a General Vision Neurosight board, and a board based on DSP TMS320c6x developed in our laboratory. We discuss first for each case the architecture of the system. Then results are presented in terms of hardware resources used and processing speed.

2.3.1 First Implementation based on FPGA

This implementation is realized on a MEMEC industrial board comprising a FPGA Xilinx SpartanII-300, which contains 3072 slices and 16 memory blocks of 512 bytes each. We have implanted on the FPGA our model of face tracking and identity verification. This implementation creates an RBF neural network with 15 hidden nodes. Each hidden node stores a center vector of 320 components. The used measured distance is the distance $d_1(x)$. The activation function of each center is a Heaviside function whose associated width delimits the influence area of the center. Figure 5 shows the organization's tasks and the coding of these tasks using VHDL description. The original video image is stored in an image memory bank with each pixel coded on a byte; the input vector extraction consists of calculating averages of four successive pixels on rows of the image. Each vector is fed to the 15 hidden nodes of the RBF network which gives their respective responses in parallel.

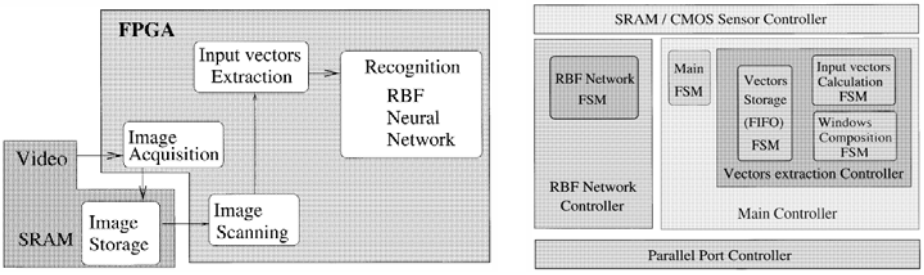

Figure 5. Organization's tasks and coding in VHDL for the first implementation

The Table 1 presents information on FPGA resources. The input vectors extraction needs 57 slices in order to define the image memory access and the interaction logic with centers. A memory block (FIFO) is necessary to store input vectors to be tested. Each trained center needs one memory block and 29 slices for calculation (distance, activation function, decision). This implementation uses 827 "slices" (27% of total resources). Note that the number of centers is limited by the number of independent internal memory blocks.

	extraction	15 centers	Interfaces & controls	Total
Number of slices used	57	435	335	827
Slices used rate	2%	14.1%	10.9%	27%
Number of Blocks RAM used	1	15	0	16
Blocks Ram used rate	6%	94%	0%	100%

Table 1. Results of the first implementation on the Memec Board

The complete implementation is realized in parallel using the pipeline technique for each stage of the processing. The images size is 288 x 352 and contains 161 x 63 = 10 143 windows of 40 x 32 pixels each with a displacement scan step along the row and the column of 2. We realized, respectively, 49.65M additions, 48.8M subtractions, 370 944 divisions, and 142 002 comparisons. The processing speed of this first implementation is 14 images per second with a success rate of 92% for face tracking and identity verification.

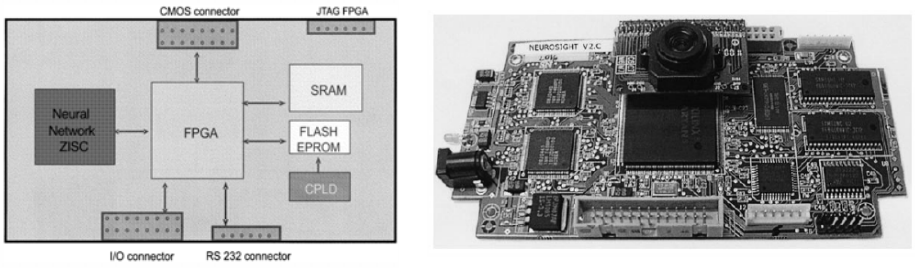

Figure 6. Neurosight block diagram and board picture

2.3.2 Second Implementation based on ZISC Chip

We also made hardware implementation of our model using a commercial board linked to pattern recognition applications. This General Vision Neurosight board contains a CMOS sensor (288 x 352 pixels), a FPGA Xilinx SpartanII-50, two memory banks of 512KB each, as well as two specific ZISC chips (see Figure 6). One ZISC chip contains 78 RBF-link nodes with a maximal length of input vectors N=64. The used measured distance and the activation function of each node are, respectively, the distance $d_1(x)$ and the Heaviside function. We adapt the complexity of the model to this embedded system. At first, we reduce the size of the original image by keeping only one line out four. This new image obtained (size 72 x 352) is then analyzed with a slippery window of 8 x 32. On each row of each window, we compute averages of eight consecutive four pixels blocks. Each window yields an input vector of 64 components to be analyzed by the ZISC chip. A total number of 10 465 windows are tested which implies 10.16 M additions, 10.05 M subtractions, 92 736 divisions, and 146 510 comparisons to be computed. We implement the input vectors extraction and all interfaces (memory access, ZISC access) on the FPGA Xilinx SpartanII. Figure 7 shows the tasks on the Neurosight board and the different levels of control coded in VHDL.

Table 2 presents information on hardware resources used for this second implementation. The input vectors extraction implementation requires the same resources as those used with the MEMEC board. Here, we use only one ZISC chip (78 nodes maximum). The processing speed of this second implementation is 25 images/s with a success rate of 85.3% for face tracking and identity verification.

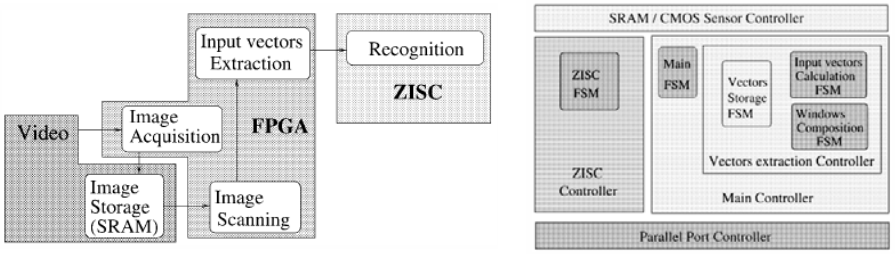

Figure 7. Organization's tasks and coding in VHDL for the second implementation

	Extraction	Interfaces & controls	Total
Total number of slices	768	768	768
Number of slices used	57	235	292
Slices used rate	7.4%	30.6%	38%
Total number of Blocks RAM	8	8	8
Number of Blocks RAM used	1	0	1
Blocks Ram used rate	12.5%	0%	12.5%

Table 2. Results of the second implementation on the Neurosight Board

2.3.3 Third Implementation based on DSP

DSPs are specific processors destined for signal and image processing. The C6x family is the last generation Texas Instruments DSP. They are available in fixed point (C62x and C64x) and floating point (C67x) versions, with CPU frequencies from 150 MHz to 1000 MHz. Our laboratory has developed a system based on a DSP TMS320 C6201B (see Figure 8). A CCD sensor sends 16-bit data to the DSP via a complex programmable logic device (CPLD). The DSP performs different processing and sends the resulting images to a PC via an USB bus. Two SDRAM memories are available to store images between the different processings.

The hardware implementation of our model for face tracking is realized on this embedded system. The goal of the implementation has been to optimize in Assembler each stage of processing using, in parallel, the maximum number of DSP functional units.

The used measured distance and the activation function of each node are, respectively, the distance d0(x) and a Gaussian function. Each vector of 320 components is fed to the 15 hidden nodes of the RBF network. The number of windows to be analyzed and the numbers of additions and divisions for input vectors extraction are the same than in the first

implementation. A correct rate of 98.2% is obtained for face tracking and identity verification.

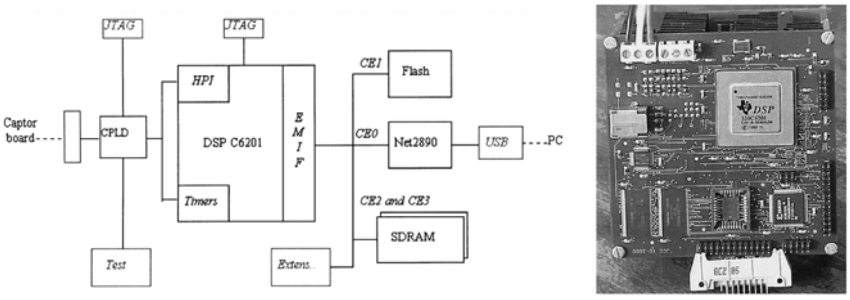

Figure 8. Block diagram and board picture of the third embedded system

Table 3 respectively shows experimental implementation results obtained using the DSP C6201 and simulation results obtained using the DSP C64x with the development tools, Code Composer Studio (Texas Instruments).

Hardware	Implementation on C6201		Simulation on C64x	
Langage	C	Assembler	C	Assembler
Input vectors Extraction	4.14 ms	1.8 ms	1.2 ms	0.14 ms
Distance calculation	211 ms	144 ms	58.8 ms	13.3 ms
Gaussian function + Decision	67 ms		22.2	
Processing speed	3.5 im. /s	4.8 im. /s	12.1 im. /s	28.6 im. /s

Table 3. Results of the third implementation on DSP

2.4 Discussion on the three Hardware implementations

We created a model that allows us to detect the presence of faces, to follow them, and to verify their identities in video sequences using a RBF neural network. The model's robustness has been tested using video sequences. The best performance has been obtained with one subsampling of a pixel/4 for each row, the measured distance $d_0(x)$ and the Gaussian activation function. In fact, the subsampling preprocessing and the application of the $d_0(x)$ distance render the model less sensitive to face details and to the small differences between training examples and test windows, thus, we have the better generalization.

We have demonstrated the feasibility of face tracking and identity verification in real time using existing commercial boards. We have implanted our model on three embedded systems. The success rate of face tracking and identity verification is, respectively, 92% (FPGA), 85% (ZISC), and 98.2% (DSP). Processing speeds obtained for images of size 288 x 352 are, respectively, 14 images/s, 25 images/s, and 4.8 images/s.

Our model integrating 15 hidden nodes allows us to distinguish two faces with a good performance (> 90% of success rate). Extending this model to recognition of more faces (> 10) necessitates a calculation power superior to 10 Giga flops and thus, new architectures must be developed. They can be developed using more effective components, for example,

FPGA Virtex 5 series or DSP TMS320C64, thus allowing a very rapid processing speed and better performance of face tracking and identity verification.

3. Design of a CMOS sensor dedicated to the extraction of input vectors

A system capable of doing face localization and recognition in real time has many applications in intelligent man-machine interfaces and in other domains such as very low bandwidth video conferencing, and video e-mail.
This section describes the main principles of a vision system, allowing to detect automatically the faces presence, to localize and to follow them in video sequences. The preliminary works, described in the previous section, have shown that RBF networks gave interesting results (Yang & Paindavoine, 2003) but imposed a fast feature extraction to reduce the size of the input vectors of the RBF network. So, the main goal of the current project is the development and the characterisation of a specific CMOS sensor able to realize the image acquisition, to extract a window of interest in the whole image and to evaluate means values of consecutive pixels on lines and columns.
A first image sensor with electronic shutter has been integrated in a 0.6 µm digital CMOS technology. The pixel cell consists of four transistors and a photodiode. Each pixel measures 30 µm by 30 µm and has a fill factor of about 40%. Each selected pixel produces a current which is transferred to the column readout amplifiers and converted by a pipeline ADC to produce a digital output. The two analog and digital values are then multiplexed to the output of the sensor. This retina also includes a logic command in order to realize acquisition of subwindows with random size and position.

3.1 Overview of the Chip Architecture

An active pixel sensor (APS) is a detector array that has at least one active transistor within the pixel unit cell (Nakamura et al., 1997). Currently, active pixel sensor technology integrates electronic signal processing and control with smart camera function onto the same single chip as a high performance image sensor (Kemeny et al., 1997). CMOS image sensors with integrated signal processing have been implemented for a number of applications (Aw & Wooley, 1996). Most current CMOS sensors have been designed for video applications, and digital photography. Improvement continues to be made because current mode image sensors have several advantages for example, low power supply, smaller place, higher operation speed (Huang & Horsney, 2003, Tabet & Horsney, 2001).
The following subsections describe the design of the image sensor using a standard 0.6 µm CMOS process. The design is based on the integration of four MOS transistors for each pixel, a column readout amplifier, a sequential control unit which includes variable input counters, decoders, multiplexers and finally an analog to digital converter. Results based on design and simulations are presented for each part of the circuit.
The architecture of the proposed image sensor is shown in Figure 9. This figure first describes the core of the system represented by the m x m array of transistors active pixels. On the left, the second block, the row decoder is charged to send to each line of pixels the control signals allowing pixel resetting, shutter opening or closing, pixel readout, ... On the bottom of the circuit, the third block is made up of amplifiers, multiplexers and column decoders whose purpose is to detect, amplify and route the signal resulting from readout column to the output of the circuit. The automatic scan of the whole array of pixels or a

subwindow of pixels is implemented by a sequential control unit which generates the internal signals to the row and column decoders. Finally, the analog output voltages are proportional to the grey scale intensity of the image. They are passed to an analog to digital converter (ADC) (as seen on the right of the block diagram). This ADC allows the conversion of analog values in digitals values which will be later processed by a DSP or a FPGA outside the chip.

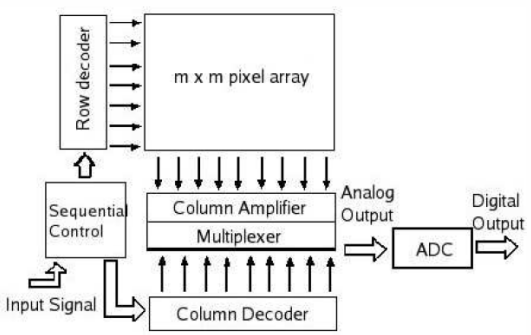

Figure 9. Image Sensor Architecture

3.2 Design of the Active Pixel Sensor

We used a standard pixel as described in the left part of Figure 2 because it is a simple and stable design (Aw & Wooley, 1996, Coulombe et al., 2000). It consists of 3 PMOS transistors, a NMOS transistor for row access and a photodiode. *m_1* is the shutter transistor, *m_2* is the reset transistor, and the transistor *m_3* acts as a transconductance buffer that converts the voltage at V_{Pix} into a current. The vertical column lines in the array are implemented using second-layer metal. First layer metal is used for the horizontal row lines. Third-layer metal is connected to V_{ss} and covers all active areas of the pixel except the photodiodes.

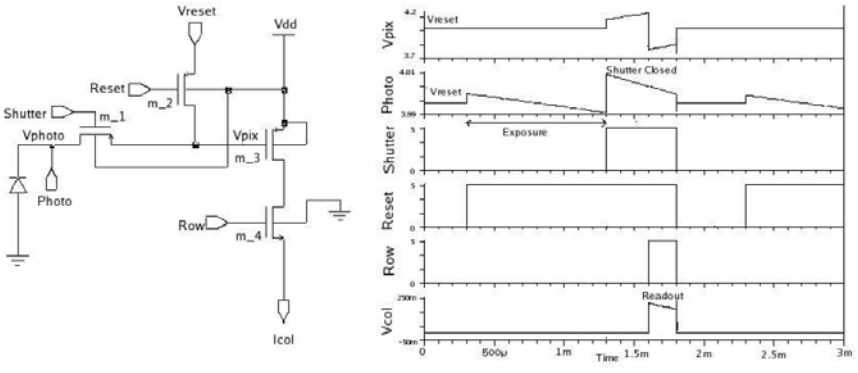

Figure 10. Pixel circuit schematic and results of simulation

Prior to the image acquisition, *m_1* and *m_2* are on, resetting node V_{Photo} and V_{Pix} to the V_{Reset} value. After reset, when *m_1* is on and *m_2* turned off, the charges generated by absorption of light are integrated onto the parasitic capacitances of the photodiode and the transistor

m_3. So, during the exposure period, voltage is accumulated at node V_{Photo} and V_{pix}. At the end of the exposure period, the shutter is closed by turning off *m_1*. Consequently, the photosignal is stored as a voltage on node V_{pix}. Finally, during readout, the row access transistor *m_4* is turned on, and the drain current of *m_3* is fed via the column line to the column readout amplifier. The right part of Figure 10 shows the main waveforms (V_{Pix}, V_{Photo}, $V_{Shutter}$, V_{Reset}, V_{Row} and V_{Col}) obtained during the simulation of one pixel. The pixels in a row are reseted by holding both reset and shutter low, turning on *m_1* and *m_2*. The voltages at nodes V_{Photo} and V_{Pix} are thereby reseted close to V_{Reset}.
During exposure, reset goes high (*m_2* turns off) while shutter is unchanged at a low value (*m_1* remains on). So, the photocurrent can be integrated onto the parasitic capacitances at V_{Photo} and V_{Pix}. At the end of the exposure period, shutter is closed by turning off *m_1* and it is cutting off the photocurrent into the node V_{Pix}. I_{Col} can be read on the column bus when *m_4* is turned on (row is high). The voltage at the drain of *m_3* falls from V_{dd} to the bias voltage of the column line, and this change couples a small negative offset into node V_{Pix}. The drain current of *m_3* is fed via the column line to the column readout amplifier.

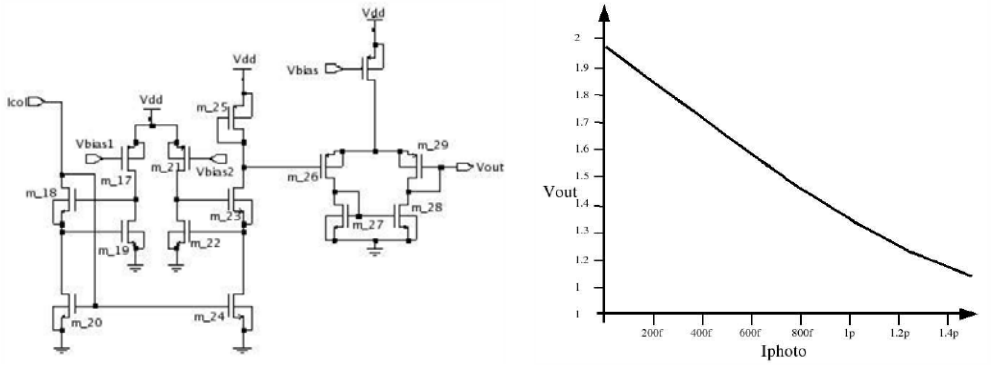

Figure 11. Column amplifier schematic and simulation results

3.3 Design of the Column Amplifier

The Fig 11 represents the electronic schematic of the column amplifier. The design of this amplifier provides a low impedance for the column lines, converts the readout current from the selected pixel into a voltage that is proportional to the integrated photovoltage in the pixel. The concept of using current mirror amplifier column is to amplify signal by duplication at the column level. Amplification is achieved by designing a current mirror *m_20* and *m_24* with ratio $W/L_{m\_20} = n \times W/L_{m\_24}$. The transistors *m_22* and *m_23* are added to enhance the output impedance of the current mirror. The circuit including *m_17*, *m_18*, *m_20* operates almost identically to a diode connected transistor, it is used to ensure that all the transistors bias voltages are matched to the output side (*m_22*, *m_23*, m_24). The transistors *m_17*, *m_21* are used to bias the feedback circuit. The transistors *m_26*, *m_27*, *m_28*, *m_29*, and *m_30* make up a differential unity gain amplifier. Once the current signal has been amplified by column current miroir amplifier, its output is suitable for any subsequent current mode image processing, either in continuous time or integration mode. In our case, these outputs will be used as inputs for the feature extracting architecture dedicated to the mean evaluation of consecutive pixels.

The pixel with its column amplifier has been simulated for a large range of photodiode currents as seen on Figure 11. The output voltages are plotted as a function of input photocurrents. Good output linearity is observed, even at very low photocurrent.

3.4 Design of the Sequential Control Unit

A framework dedicated to the sequential readout of successive rows and columns has been designed. The system offers the availability to program the location and the size of any window of interest in the whole image. Advantages of a such technology are large: random access of any pixel or subwindow, increase of acquisition frequency, ... In our main goal of face tracking, these aspects are crucial because only windows of interest will be scanned by the sensor. Each line of pixels included in the subwindow follows the same sequence of reading but at different moments in order to multiplex the outputs. As seen previously, each pixel is controlled by 3 signals: reset, shutter, and select. The Figure 12 shows the readout sequence of 2 successive rows.

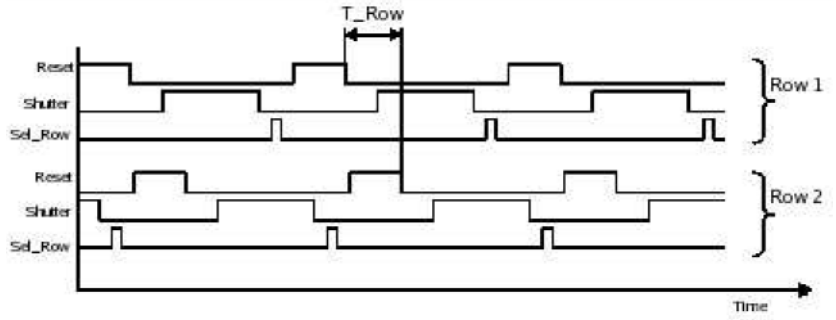

Figure 12. Timing diagram of the rows control signals

To implement the sequential control, we need counters with variable inputs: the first one for the starting position of the subwindow and the second one for its ending position. Our design is inspired by a 74HC163 counter from Philips Semiconductors. This circuit starts counting from a value which can be freely selected. It has been modified in order to add the second input corresponding to the stop value of the counting process.
Associated with the counters, the control unit uses row decoders to active the pixels rows. The row decoder is adopted from (Baker et al., 1998). A long L MOS transistor is used to pull low the output of the decoder when that particular output is not selected. The result is that all decoder outputs are zero except for the output that is selected by the input address. Two inverters are used to drive the word line capacitance. Finally, a multiplexer is used to select and pass output voltages from the column amplifiers. We use a simple design based on pairs of transistors Nmos and Pmos.

3.5 Design of the Analog to Digital Converter

Most designs of video-rate analog to digital converters (ADC's) of 8 bit resolution are implemented through flash architectures and bipolar technologies (Lewis et al., 1992). In recent years, pipelined switched capacitor topologies have emerged as an approach to implement power efficient nyquist-rate ADCs that have medium-to-high resolution (10-13 bits) at medium-to high conversion rates (Thomson & Wooley, 2001). Here, we present a 8

bit ADC operating at a 5 V supply that achieves a sample rate of about 20 Msamples/s. An experimental prototype of this converter has been implemented in a 0.6 μm CMOS process.

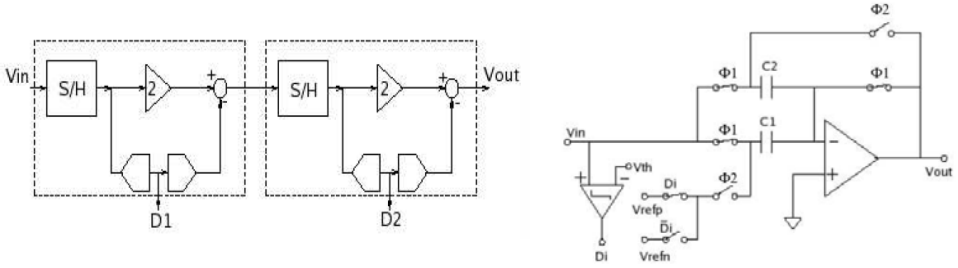

Figure 13. Pipeline ADC Architecture and Associated Circuit

Figure 13 shows the block diagram of a 1-bit per stage pipelined A/D converter. The pipelined ADC consists of N stages connected in series; two stages are only shown on the Figure 13. Each stage contains a sample and hold (S/H), a comparator, a subtractor and an amplifier with a gain of two. The pipelined ADC is an N-step converter, with 1 bit being converted per stage. The most significant bits are resolved by the first stages in the pipeline. The result of each stage is passed to the next stage in which the cycle is repeated. A pipeline stage is implemented by the conventional switched capacitor (Sonkusale et al., 2001) as shown in the Figure 13. Each stage consists of two capacitors C1 and C2 for which the values are nominally identical, an operational amplifier and a comparator. Each stage operates in two phases: a sampling phase and a multiplying phase. During the sampling phase ϕ_1, the comparator produces a digital output D_i. D_i is equal to 1 if $V_{in} > V_{th}$ and D_i is 0 if $V_{in} < V_{th}$, where V_{th} is the threshold voltage defined as the mean value between V_{refp} and V_{refn}. V_{refp} is defined as the positive reference voltage and V_{refn} as a negative reference voltage. During the multiplying phase, C_2 is connected to the output of the operational amplifier and C_1 is connected to either the reference voltage V_{refp} or V_{refn}, depending on the bit value D_i. If D_i = 1, C_1 is connected to V_{refp}, resulting in the following remainder $V_{out}(i) = 2\ V_{in}(i) - D_i V_{refp}$. Otherwise, C_1 is connected to V_{refn}, giving an output voltage $V_{out}(i) = 2\ V_{in}(i) - \overline{D_i}\ V_{refn}$.

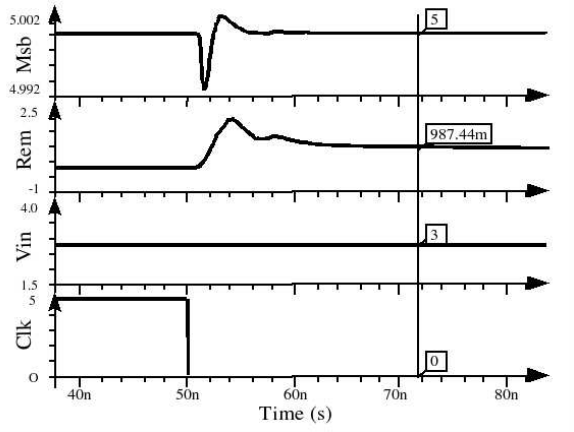

Figure 14. Simulation of one stage A/D converter

The simulation of one stage A/D converter can be seen on the Figure 14 on which the computed bit, the remainder, the input value and the clock are presented from top to bottom. The input value is V_{in} = 3V involving the output bit D_i obtains a high value. The remainder is then evaluated as the difference between $2V_{in}$ and V_{refp} (ie 2 * 3 - 5 = 1V).

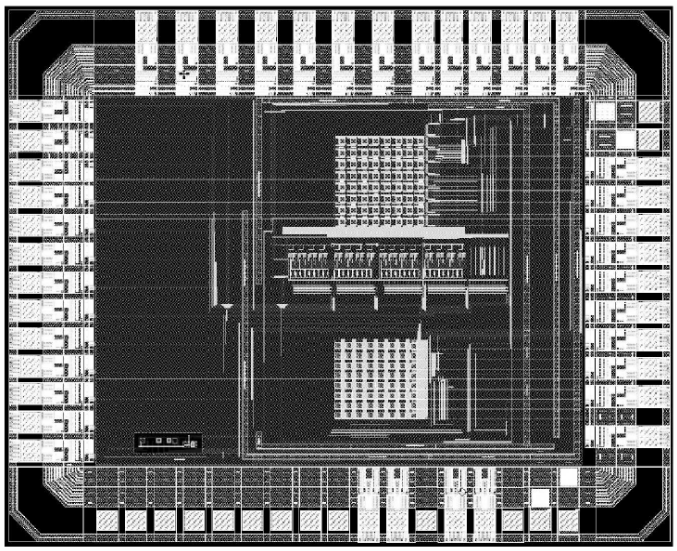

Figure 15. Layout of the test chip

3.6 Preliminary results

We have presented here results from simulations intended to evaluate and validate the efficiency of our approach. Every element described in these sections has been designed on a standard 0.6 µm CMOS Process. Two test circuits have been sent in foundry to be fabricated in 2004 and 2005. Unfortunately, the first circuit has some bugs in the design of analog output multiplexer preventing any measure. The second circuit (see Figure 15) includes any of the individual structures depicted in the previous sections of this chapter, except the ADC. So, every structure has been validated by experimental measures, showing the validity of the concepts embedded in the chip design.

Actual work focuses on the last part of the sensor ie the development of the feature extracting architecture dedicated to the mean evaluation of consecutive pixels. For this purpose, two main approaches are envisaged. First, the mean values of 4 consecutive pixels can be digitally computed and takes place after the ADC in the chip. This can be done by an adder of four 8-bit words producing a 10-bit result. The average of the four values can be easily extracted on the 8 MSB (Most Significant Bits) of the results. Second, the evaluation of the mean values can be made with the analog signals going out the column amplifiers. A dedicated circuit must take place between the column amplifiers and the ADC. Our main short-term perspective is to explore these two potential solutions, to design the corresponding chips and to evaluate their performances.

The next effort will be the fabrication of a real chip in a modern process such as a 130 nm CMOS technology. The main objective will be the design of a 256 x 256 pixel array with a pixel size of less than 10 μm x 10 μm. This chip will include all the necessary electronics allowing the extraction of parameters which can serve as inputs of a RBF neural network dedicated to face recognition.

4. Development of a fast panoramic face mosaicking and recognition system

Biometry is currently a very active area of research, which comprises several subdisciplines such as image processing, pattern recognition, and computer vision (Kung et al., 2005). The main goal of biometry is to build systems that can identify people from some observable characteristics such as their faces, fingerprints. Faces seem to have a particularly strong appeal for human users, in part because we routinely use facial information to recognize each other. Different techniques have been used to process faces such as neural network approaches (Howel & Buxton, 1998) eigenfaces (Turk & Pentland, 1991) and Markov chains (Slimane et al., 1999) As the recent DARPA-sponsored vendor test showed, most systems use frontal facial images as their input patterns (Phillips et al., 2003) As a consequence, most of these methods are sensitive to pose and lighting conditions. One way to override these limitations is to combine modalities (color, depth, 3-D facial surface, etc.) (Tsalakanidou et al., 2003, Hehser et al., 2003, Bowyer et al., 2004).
Most 3-D acquisition systems use professional devices such as a travelling camera or a 3-D scanner (Hehser et al., 2003, Lu et al., 2004). Typically, these systems require that the subject remain immobile during several seconds in order to obtain a 3-D scan, and therefore these systems may not be appropriate for some applications, such as human-expression categorization using movement estimation, or real-time applications. Also, their cost can easily make these systems prohibitive for routine applications. In order to avoid using expensive and time-intensive 3-D acquisition devices, some face recognition systems generate 3-D information from stereo vision (Wang et al., 2003). Complex calculations, however, are needed in order to perform the required self-calibration and 2-D projective transformation (Hartly et al., 2003). Another possible approach is to derive some 3-D information from a set of face images, but without trying to reconstitute the complete 3-D structure of the face (Tsalakanidou et al., 2003).
For the last ten years, our laboratory has worked on face processing and obtained results for 2-D face tracking and recognition. The goal of the present section is to describe a system that is simple and efficient and that also can potentially process 3-D faces in real time. Our method creates panoramic face mosaics, which give some 3-D surface information. The acquisition system is composed of five cameras, which together can obtain simultaneously five different views of a given face. One of its main advantages is easy setup and very low cost. This section is organized as follows. First, we describe our acquisition system. Then, we describe the method for creating panoramic face mosaics using successive linear transformations. Next, we present experimental results on panoramic face recognition. Finally, we conclude and explore possible follow-ups and improvements.

4.1 Acquisition system

Our acquisition system is composed of five Logitech 4000 USB cameras with a maximal resolution of 640 x 480 pixels. The parameters of each camera can be adjusted

independently. Each camera is fixed on a height adjustable sliding support in order to adapt the camera position to each individual (see Figure 16). The acquisition program grabs images from the five cameras simultaneous (see Figure 16). These five images are stored in the PC with a frame data rate of 20 x 5 = 100 images per second.

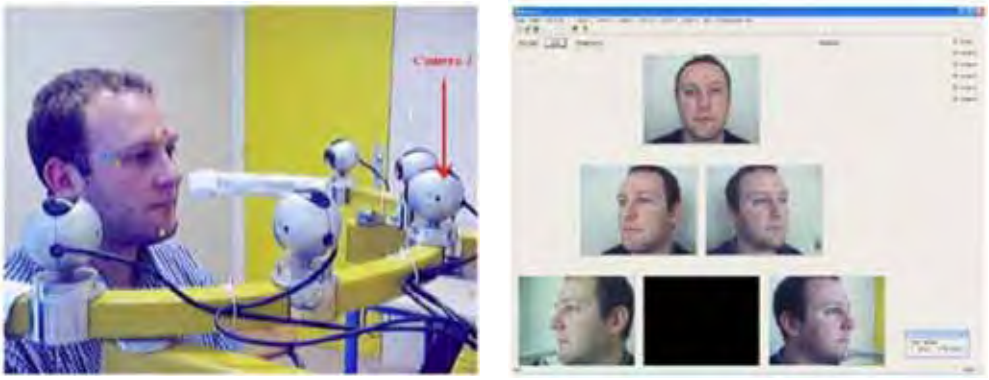

Figure 16. Acquisition system with 5 cameras and example of 5 images collected from a subject

The human subject sits in front of the acquisition system, directly facing the central camera (camera 3). Different color markers are placed on the subject's face. These markers are used later on to define common points between different face views. The positions of these color markers correspond roughly to the face fiduciary points. There are ten markers on each face, with at least three markers in common between each pair of face views.

4.2 Panoramic Face Construction

Several panoramic image construction algorithms have been already introduced. For example, Jain & Ross (2002) have developed an image-mosaicking technique that constructs a more complete fingerprint template using two impressions of the same finger. In their algorithm, they initially aligned the two impressions using the corresponding minutiae points. Then, this alignment was used by a modified version of the iterative closest point (ICP) algorithm in order to compute a transformation matrix that defines the spatial relationship between the two impressions. A resulting composite image is generated using the transformation matrix, which has six independent parameters: three rotation angles and three translation components about the x, y, and z axes.

For faces, Liu & Chen (2003) have proposed using facial geometry in order to improve the face mosaicking result. They used a spherical projection because it works better with the head motion in both horizontal and vertical directions. They developed a geometric matching algorithm in order to describe the correspondences between the 2-D image plane space QUV and the spherical surface space $O\alpha\beta$.

In general, the methods using nonlinear transformations and iterative algorithms obtain very correct results in terms of geometric precision. However, these methods require a large number of computations and therefore cannot be easily implemented in real time. Because ultimately we want to be able to build a real-time system, we decided to use simple (and therefore fast) linear methods. Our panoramic face construction algorithm is performed in

three stages: (1) marker detection and marker coordinate calculation, (2) transformation matrix estimation and image linear transformation, and (3) creation of panoramic face mosaics.

4.2.1 Marker Detection and Marker Coordinate Calculation

The first step of the algorithm corresponds to the detection of the markers put on the subject's face. The markers were made of adhesive paper (so that they would stick to the subject's face). We used three colors to create ten markers (four blue, three yellow, and three violet ones). In order to detect the markers, we used color segmentation based on the hue and saturation components of each image. This procedure allows strong color selectivity and small sensitivity to luminosity variation. First, color segmentation gives, from the original image a binary image that contains the detected markers. Then, in order to find the marker coordinates, we used a logical AND operation, which was performed between the binary image and a grid including white pixels separated by a fixed distance. This distance was chosen in relation to the marker area. A distance of 3 pixels allows us to capture all white zones (detected markers). Finally, we computed the centers of the detected zones. These centers give the coordinates of the markers in the image.

4.2.2 Transformation-Matrix Estimation and Image Linear Transformation

We decided to represent each face as a mosaic. A mosaic face is a face made by concatenation of the different views pasted together as if they were on a flat surface. So, in order to create a panoramic face we combine the five different views. We start with the central view and paste the lateral views one at a time. Our method consists of transforming the image to be pasted in order to link common points between it and the target image. We obtain this transformed image by multiplying it by a linear transformation matrix T. This matrix is calculated as a function of the coordinates of three common markers between the two images. C1 and C2 represent, respectively, the coordinates of the first and second images:

$$C_1 = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \qquad C_2 = \begin{bmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \end{bmatrix} \qquad \mathrm{T} = \mathrm{C}_1 \times (C_2^*)^{-1}$$
$$with\ C_2^* = \begin{bmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{bmatrix} \quad and \quad T = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \tag{4}$$

Then, we generalize this transformation to the whole image: $x = a_1x' + b_1y' + c_1$ and $y = a_2x' + b_2y' + c_2$. This linear transformation corresponds to a combination of image rotation, image translation, and image dilation. The two first images on Figure 17 represent an example of the linear transformation on the image 4. The right part of the figure depicts the superposition of image 3 (not transformed) and image 4 (transformed).

4.2.3 Creation of Panoramic Face Mosaics

We begin the panoramic face construction with the central view (image 3). From the superposition of the original image 3 and transformed image 4 (see Figure 17), we remove redundant pixels in order to obtain a temporary panoramic image 3-4 (see Figure 18, first

image). In order to eliminate redundant pixels, we create a cutting line that goes through two yellow markers. This panoramic image 3-4 temporarily becomes our target image. We repeat this operation for each view. First, image 2 is pasted on the temporary panoramic image 3-4 in order to obtain a new temporary panoramic image 2-3-4 (see Figure 18, second image). The corresponding transformation matrix is generated using three common violet markers. Then, we compute the transformation matrix that constructs image 2-3-4-5 (see Figure 18, third image) using two blue markers and one yellow marker. Finally, image 1 is pasted to the temporary panoramic image 2-3-4-5 with the help of two blue markers and one violet marker (see Figure 18, fourth image).

Figure 17. From left to right, Image 4 before and after the linear transformation, original image 3 and superposition of transformed image 4 and original image 3

Figure 18. Mosaicking results: image 3-4, image 2-3-4, image 2-3-4-5, and , image 1-2-3-4-5

Figure 19 displays some examples of the final panoramic face composition from five views. This composition preserves some of the face shape. For example, the chin of a human face possesses more curvature than other parts; therefore the bottom part of the panoramic face is composed of five views: 1, 2, 3, 4, and 5. On the other hand, three views (1, 3, and 5) suffice to compose the top part. Figure 19 shows final mosaic faces obtained after automatic contour cutting. For this, we first surround the panoramic face with a circle that passes by the extreme points of the ears in order to eliminate the background. Then, we replace segments of this circle by polynomial curves using extreme-point coordinates located with the help of the marker positions. Note that these ten markers allow us to link common points between five views. The coordinates of the markers are computed in the marker detection process and arranged in a table. Then, all ten markers are erased from all five views, using a simple image-processing technique (local smoothing). This table of marker coordinates is regenerated for each temporary panoramic image construction. The goal of marker elimination is to use panoramic faces for face recognition or 3-D face reconstruction.
As compared to the method proposed by Liu & Chen (2003) panoramic faces obtained using our model are less precise in geometry. For example, Liu and Chen used a triangle mesh in order to represent a face. Each triangle possesses its own transformation parameters. In our system, a single transformation matrix is generated for a complete image. Liu and Chen have also established a statistical modeling containing the mean image and a number of "eigenimages" in order to represent the face mosaic. Our objective is to study an efficient and simple algorithm for later hardware implantations. Methods necessitating a large

calculation volume and a large memory space are not adapted to embedded systems. In order to test and validate our panoramic face mosaicking algorithm, we propose, in the next sections, a study of face recognition based on the eigenface model proposed by Turk & Pentland (1991). With our method, we created a panoramic face database composed of 12 persons x 4 expressions x 2 sessions = 96 panoramic faces. The two acquisition sessions were performed over an interval of one month. The four expressions were: neutral, smile, deepened eyebrows, and eyes closed (see Figure 19). We implemented a face recognition procedure using this database.

Figure 19. Examples of panoramic faces

4.3 Face Recognition Description

Over the past 25 years, several face recognition techniques have been proposed, motivated by the increasing number of real-world applications and also by the interest in modelling human cognition. One of the most versatile approaches is derived from the statistical technique called principal component analysis (PCA) adapted to face images (Valentin et al., 1994). Such a approach has been used, for example, by Abdi (1988) and Turk & Pentland (1991) for face detection and identification. PCA is based on the idea that face recognition can be accomplished with a small set of features that best approximates the set of known facial images. Application of PCA for face recognition proceeds by first performing a PCA on a well-defined set of images of known human faces. From this analysis, a set of K principal components is obtained, and the projection of the new faces on these components is used to compute distances between new faces and old faces. These distances, in turn, are used to make predictions about the new faces. Technically, PCA on face images proceeds as follows. The K face images to be learned are represented by K vectors a_k, where k is the image number. Each vector a_k is obtained by concatenating the rows of the matrix storing the pixel values (here, gray levels) of the k'th face image. This operation is performed using the vec operation, which transforms a matrix into a vector (see Abdi et al. (1995) for more details).

The complete set of patterns is represented by a I x K matrix noted A, where I represents the number of pixels of the face images and K the total number of images under consideration. Specifically, the learned matrix A can be expressed as $A = P \Delta Q^T$ where P is the matrix of eigenvectors of AA^T, Q is the matrix of eigenvectors of A^TA, and Δ is the diagonal matrix of singular values of A, that is, $\Delta=\Lambda^{1/2}$, with Λ, the matrix of eigenvalues of AA^T and A^TA. The left singular eigenvectors P can be rearranged in order to be displayed as images. In general, these images are somewhat facelike (Abdi, 1988) and they are often called eigenfaces. Given the singular vectors P, every face in the database can be represented as a weight vector in the principal component space. The weights are obtained by projecting the face image onto the left singular vectors, and this is achieved by a simple inner product operation: $PROJ_x=X^TP^{\Delta-1}$ where x is a facial vector, corresponding to an example face in the training process or a test face in the recognition process. Therefore, when a new test image whose identification is required is given, its vector of weights also represents the new image.
Identification of the test image is done by locating the image in the known face database whose weights have the smallest Euclidean distance from the weight of the test image. This algorithm, employed by Turk and Pentland is called the nearest neighbor classification rule.

4.4 Experimental results on Panoramic Face Recognition

For these first tests, panoramic faces were analyzed using the original 240x320-pixel image (spatial representation) without preprocessing. The database consisted of 12 persons x 4 expressions x 2 1sessions = 96 panoramic faces, and was divided into two subsets. One subset served as the training set, and the other subset as the testing set. As illustrated in Figure 19, all these panoramic faces possess a uniform background, and the ambient lighting varied according to the daylight.
From the panoramic face database, one, two, three, or four images were randomly chosen for each individual in order to create the training set (number of patterns for learning per individual, p=1, 2, 3, 4). The rest of the panoramic faces were used in order to test the face recognition method. For example, when p=1, the total number of training examples is equal to 1 x 12 persons = 12, and the number of test samples for recognition is equal to 96−12=84. Therefore, for each individual, only one panoramic face is learned in order to recognize seven other images of this person. Several executions of our MATLAB program were run for each value of p, using randomly chosen training and testing sets. Then we computed the mean performance. Using the nearest neighbour classification rule, the panoramic face identity test is done by locating the closest image in the known face database. Therefore, the system can make only confusion errors (i.e., associating the face of one person with a test face of another). Correct panoramic face recognition rates go from 70 % when p=1 to 93.2% when p=4.
We added a discriminant analysis stage in the face recognition process so as to determine the number of necessary eigenvectors. This analysis, called the jackknife (Yang & Robinson, 2001) reorders eigenvectors, not according to their eigenvalues, but according to their importance for identification. Specifically, we computed the ratio of the between-group inertia to the within-group inertia for each eigenvector. This ratio expresses the quality of the separation of the identity of the subject performed by this eigenvector. The eigenvector with the largest ratio performs the best identity separation, the eigenvector with the second largest ratio performs second best, etc. We observe that it suffices to use only 23

eigenvectors to reach the maximum recognition rate (93.2%). Additional eigenvectors do not add to the quality of the identification.
We also tested the frequential behavior of our recognition system. We can observe that the frequential spectra of a panoramic face are well centered at low frequencies. This allows us to apply a lowpass filter in order to reduce the size of the data set to process. Only 80 x 80 FFT amplitude values of low frequencies were used for the recognition system.

Figure 20. (a) original image, (b) original image with added Gaussian noise, (c) FFT image using the spectrum amplitude of (b) and the phase of (a) and (d) FFT image using the spectrum amplitude of (a) and the phase of (b)

We applied the same training and testing process as used in spatial representation. We obtain a better recognition rate with the frequential representation (97.5%) than with the spatial representation (93.2%). This advantage of the frequential representation is due to the fact that for face images, the spectrum amplitude is less sensitive to noise than the spectrum phase. We confirmed this interpretation by using a panoramic face image to which noise was added. Figure 20(a) shows a original panoramic face. Figure 20 (b) displays the same panoramic face image with added noise. We first obtained the FFTs of these two images and then their inverse FFTs in the two following manners: (1) using the spectrum amplitude of the noised image and the spectrum phase of the original image (see Figure 20-c) and (2) using the spectrum phase of the noised image and the spectrum amplitude of the original image (see Figure 20-d).
These results show that the face obtained with the first configuration is closer to the original face than the face obtained with the second configuration. This confirms that the spectrum amplitude is less sensitive to noise than the spectrum phase.

4.5 Panoramic face recognition with negative samples

In order to evaluate the behavior of our system for unknown people, we added four people to the test database. These panoramic faces were obtained as described in Sec. 4.2. Table 4 displays the performance of different tests. In order to reject these unknown faces, we established a threshold of Euclidean distance. Because we are working on applications of typical access control, where confusion is more harmful than nonrecognition, we decided to use a severe acceptance threshold in order to reject intruders. Note that the acceptance threshold is constant for all tests. Efficiency is defined as follows:

- Recognition: Correct recognition of a panoramic face.
- Nonrecognition: A panoramic face has not been recognized.
- Confusion: A panoramic face is confused with an intruder.

These performance results are obtained using the frequential representation and show that performance declines in comparison with tests without negative samples.

Number of training examples per individual p	Total number of training examples	Number of eigenvectors used	Number of tests for recognition	Non recognition rate (%)	Confusion rate (%)	Recognition rate (%)
1	12	8	116	25.4	5.85	68.75
2	24	13	104	12.74	4	83.26
3	36	18	92	7.58	3.5	88.92
4	48	24	80	4.82	2.8	92.38

Table 4. Results of panoramic face recognition with negative samples

4.6 Discussion

In this section, we have proposed a fast and simple method for panoramic face mosaicking. The acquisition system consists of several cameras followed by a series of fast linear transformations of the images. The simplicity of the computations makes it possible to envisage real-time applications.

In order to test the recognition performance of our system, we used the panoramic faces as input to a recognition system based on PCA. We tested two panoramic face representations: spatial and frequential. We found that a frequential representation gives the better performance, with a correct recognition rate of 97.46%, versus 93.21% for spatial representation. An additional advantage of the frequential representation is that it reduces the data volume to be processed and this further accelerates the calculation speed. We used negative samples for the panoramic face recognition system, and the correct recognition rate was 92.38%.Experimental results show that our fast mosaicking system provides relevant 3-D facial surface information for recognition application. The obtained performance is very close or superior to published levels (Howell & Buxton, 1998, Slimane et al., 1999, Tsalakanidou et al., 2003).

In the future, we plan to simplify our acquisition system by replacing the markers with a structured light. We also hope to use our system without markers. For this, we will detect control points on faces (corners, points of maximum curvature, etc.). Another line of development is to improve the geometry quality of our panoramic face mosaic construction (Liu & Chen, 2003, Puech et al., 2001). For this, we will use realistic human face models. We are also exploring processing panoramic face recognition using other classifiers with more variable conditions.

5. Conclusions

In this chapter, we have presented three dedicated systems to face recognition developed by our research team since 2002. Our main objective was motivated by the implementation on embedded systems of efficient models of unconstrained face tracking and identity verification in arbitrary scenes. The main goal of these various systems is to provide efficient algorithms that only require few hardware in order to obtain high success rates of face recognition with high real time constraints.

The first system is a real time vision system that allows us to localize faces in video sequences and verify their identity. These processes are image processing techniques and

the radial basis function (RBF) neural network approach. The robustness of this system has been evaluated quantitatively on eight video sequences. We have also described three hardware implementations of our model on embedded systems based, respectively, on field programmable gate array (FPGA), zero instruction set computer (ZISC) chips, and digital signal processor (DSP). For each configuration, we have analyzed the algorithm complexity and present results of implementations in terms of resources and processing speed.
The main results of these first implementations have highlighted the need of a dedicated hardware such as an artificial retina embedding low level image processing in order to extract input vectors of the RBF neural network. Such a system could unload a consequent calculation part of FPGA. So, the second part of the chapter was devoted to the description of the principles of an adequate CMOS sensor. For this purpose, a current mode CMOS active sensor has been designed using an array of pixels that are amplified by using current mirrors of column amplifiers. This circuit is simulated using Mentor Graphics™software with parameters of a 0.6 µm CMOS process. The circuit is able to realise captures of subwindows at any location and any size in the whole image and computes mean values of adjacent pixels which can serve as inputs of the RBF network.
In the last section of this chapter, we present some new results on a system that performs mosaicking of panoramic faces. Our objective was to study the feasibility of panoramic face construction in real time. We built a simple acquisition system composed of five standard cameras, which together can take simultaneously five views of a face at different angles. Then, we chose an easily hardware-achievable algorithm, consisting of successive linear transformations, in order to compose a panoramic face from these five views. In order to validate our system, we also conducted a preliminary study on panoramic face recognition, based on the principal-component method. Experimental results show the feasibility and viability of our system and allow us to envisage later a hardware implementation.

6. References

Abdi, H. (1988). A generalized approach for connectionist auto-associative memories: interpretation, implications and illustration for face processing, in *Artificial Intelligence and Cognitive Sciences*, J. Demongeot (Ed.), Manchester Univ. Press.

Abdi, H.; Valentin, D.; Edelman, B. & O'Toole, A. (1995). More about the difference between men and women: evidence from linear neural networks and the principal component approach, *Perception* vol 24, pp 539–562.

Aw, C. & Wooley, B. (1996) .A 128x128 pixel standard cmos image sensor with electronic shutter,. *IEEE Journal of Solid State Circuits*, vol. 31, no. 12, pp. 1922-1930.

Baker, R.; Li, H.; & Boyce; D. (1998). *CMOS Circuit Design Layout and Simulation*. New-York: IEEE Press.

Bowyer, K.; Chang, K. & Flynn, P. (2004). A survey of 3D and multi- modal 3*D*+2*D* face recognition, in *Proc. Int. Conf. on Pattern Recognition (ICPR)*.

Buhmann, J.; Lades, M. & Eeckmann, F. (1994). Illumination-invariant face recognition with a contrast sensitive silicon retina, in *Advances in Neural Information Processing Systems (NIPS)*. New York: Morgan Kaufmann, 1994, vol. 6, pp. 769–776.

Coulombe, J.; Sawan, M. & Wang, C. (2000).Variable resolution CMOS current mode active pixel sensor. in *IEEE International Symposium on Circuits and Systems*.

Férand, R.et al. (2001). A fast and accurate face detector based on neural networks, *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 23, pp. 42–53.

Hartly, R. & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge Univ. Press

Hehser, C.; Srivastava, A. & Erlebacher, G. (2003). A novel technique for face recognition using range imaging, in *Proc. 7th Int. Symp. On Signal Processing and Its Applications.*

Howell, A. & Buxton, H. (1998). Learning identity with radial basis function networks, *Neurocomput.*, vol. 20, pp. 15–34.

Huang, Y. & Horsney, R. (2003). Current-mode cmos image sensor using lateral bipolar phototransistors,. *IEEE Trans. on Electron Devices*, vol. 50, no. 12, pp. 2570-2573.

Jain, A. & Ross, A. (2002). Fingerprint mosaicking, in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

Kemeny, S.; Mathias, L. & Fossum, E. (1997). Multiresolution image sensor, *IEEE Transactions on Circuit and System for Video Technology*, vol. 7, no. 4, pp. 575-583.

Kobayashi, K. (2001). Mobile terminals and devices technology for the 21st century, Spec. *Rev. Paper – New Start 21st Century: NEC Res. Develop.*, vol. 42, no. 1.

Koh, L.; Ranganath, S. & Venkatesh, Y. (2002). An integrated automatic face detection and recognition system, *Pattern Recogn.*, vol. 35, pp. 1259–1273.

Kung, S.; Mak, M. & Lin S. (2005), *Biometric Authentification: A Machine Learning Approach*, Prentice-Hall, Upper Saddle River, NJ

Lewis, S.; Fetterman, H.; Gross, G.; Ramachandran, R. & Viswanathan T. (1992), .10-b 20-Msample/s analog-to-digital converter. *IEEE Journal on Solid-States Circuits*, vol. 27, no. 3, pp. 351-358.

Lindblad, T.et al. (1995). Implementating of the new zero instruction set computer (ZISC036) from IBM for a Higgs Search, *Nucl. Instrum. Methods*, vol. A357.

Liu, H. (1998). An automatic human face recognition system, *Opt. Lasers Eng.*, vol. 30, pp. 305–314.

Liu, X. & Chen, T. (2003). Geometry-assisted statistical modeling for face mosaicing, in IEEE Int. Conf. on Image Processing (ICIP), Vol. 2, pp.883–886.

Lu, X.; Colbry, D. & Jain, A. (2004). Three-dimensional model based face recognition, in *Proc. Int. Conf. on Pattern Recognition*, pp. 362–366.

Matsumoto, Y. & Zelinsky, A. (2000). An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement, in Proc. *4th IEEE Int. Conf. Automatic Face and Gesture Recognition*, Grenoble, France, pp. 499–504.

Nakamura, J.; Pain, B; Nomoto, T.; Nakamura, T. & E. Fossum. (1997). Onfocal plane signal processing for current-mode active pixel sensors. *IEEE Transaction on Electrons Devices*, vol. 44, no. 10, pp. 1747-1758.

Park, I. & Sandberg, I. (1991). Universal approximation using radial basis function networks, Neural Computat., vol. 3, pp. 246–257.

Pérez-Uribe, A. & Sanchez, E. (1996). FPGA implementation of an adaptable- size neural network, in Proc. *Int. Conf. Artificial Neural Networks ICANN'96*, Bochum, Germany.

Phillips, P.; Grother, P.; Micheals, R.; Blackburn, D.; Tabassi, E. & Bone, J. (2003). Face recognition vendor test 2002, presented at *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG).*

Puech, W.; Bors, A.; Pitas, I. & Chassery, J.-M. (2001). Projection distortion analysis for flattened image mosaicking from straight uniform generalized cylinders, *Pattern Recogn.* vol 34, pp 1657–1670.

Rosenblum, M.; Yacoob, Y. & Davis, L. (1996). Human expression recognition from motion using a radial basis function network architecture, *IEEE Trans. Neural Networks*, vol. 7, pp. 1121–1138.

Rowley, H.; Baluja, S. & Kanade, T. (1998). Neural network-based face detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 23–38.

Sim, T.et al. (2000). Memory-based face recognition for visitor identification, in Proc. *4th IEEE Int. Conf. Automatic Face and Gesture Recognition*, Grenoble, France, pp. 26–30.

Skrbek, M. (1999). Fast neural network implementation, *Neural Network World*, vol. 5, pp. 375–391.

Slimane, M.; Brouard, T.; Venturini, G. & Asselin de Beauville, J. P. (1999). Unsupervised learning of pictures by genetic hybridization of hidden Markov chain, *Signal Process.* vol 16, no 6, 461–475.

Sonkusale, S.; Speigel, J. & Nagaraj, K. (2001) .Background digital error correction technique for pipelined ADC,. *In Proc of IEEE ISCAS*, vol 1, pp 408-411.

Tabet, M. & Hornsey, R. (2001). Cmos image sensor camera with focal plane edge detection,. in *Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 1129-1133.

Thomson, D. & Wooley, B. (2001).A 15-b pipelined cmos _oating point a/d converter,. *IEEE Journal on Solid-States Circuits*, vol. 36, no. 2, pp. 299-303.

Tsalakanidou, F.; Tzovaras, D. & Strintzis, M. (2003). Use of depth and colour eigenfaces for face recognition, *Pattern Recogn. Lett.* vol 24, pp 1427–1435.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition, *J. Cogn Neurosci,.* vol 3, pp 71–86.

Valentin, D.; Abdi, H.; O'Toole, A. & Cottrell, G. (1994). Connectionist models of face processing: a survey, *Pattern Recogn.* vol 27, pp 1208–1230.

Wang, J.; Venkateswarlu, R. & Lim, E. (2003). Face tracking and recognition from stereo sequence, *Comput. Sci.* vol 2688, pp 145–153.

Yang, F. & Paindavoine, M. (2003) , Implementation of a RBF neural network on embedded systems: real time face tracking and identity verification, *IEEE Trans. Neural Netw.* vol 14, no 5, pp 1162–1175.

Yang, M.; Kriegman, D. & Ahuja, N. (2001). Detecting faces in images: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 42–53.

Yang, M. & Robinson D. (1986). *Understanding and Learning Statistics by Computer*, World Scientific, Singapore.

Yeh, Y. & Lee, C. (1999). Cost effective VLSI architectures and buffer size optimization for full search block matching algorithms, *IEEE Trans. VLSI Syst.*, vol. 7, pp. 345–358,