

EFFECTIVE VIDEO PROCESSING ARCHITECTURES FOR FACE RECOGNITION ALGORITHMS IMPLEMENTATION

Radu ARSINTE

Technical University Cluj-Napoca, Communication Department, Phone: +40-264-595699, Str. Baritiu 26-28, Radu.Arsinte@com.utcluj.ro

Abstract: The paper presents a study of autonomous face recognition systems based on high performance DSP, so called Media Processors and on Field Programmable Gate Array (FPGA) devices. An overview of the most powerful Media Processors and FPGAs available today is followed by an analysis of the common features of the face recognition algorithms used to implement an application. The paper presents also a generic system implemented using DSP Media processors, both from a hardware or software perspective. The presentation is focused on the possibility of implementation for face recognition algorithms in embedded systems, taking as examples databases with a limited number of faces. Results, estimations and benchmarks for the generic systems are also presented at the end of the paper.

Key words: Video processing, Media Processors, FPGA, DSP.

I. VIDEO ACQUISITION ARCHITECTURES

Biometric security systems and video surveillance are two important fields, attracting the interest both of industry and scientific community. An important task in these systems is the face recognition, allowing an identification of a person using the image of his face. The research in this field is over 30 years old, but a more intense activity is seen after 1990. This research led to efficient algorithms and commercial applications based on such algorithms. Transferring the algorithms from laboratory to real applications, makes possible to emphasize apparently minor performances, like energy consumption or overall cost, became essential. This class of applications (embedded) is possible to be implemented using high-performance components like Media Processors or FPGA based solutions.

1. Media processors

Media processors based systems could choose over a large basis of different devices. We have chosen two representative devices for this kind of processors: Texas Instruments TMS320C64xx and Philips Nexperia.

The two families are important and with a bright perspective, taking into account the tradition of the two companies in DSP (for Texas) or video processing (for Philips).

TMS320C64xx - Texas Instruments

The TMS320DM64x generation of digital media processors offers multi-channel encoding, decoding and transcoding for all of free standards and proprietary video and audio algorithms at a variety of price and performance points. Multiple performance options, multiple price options and multiple integration options are now available to you.

These fully software-programmable devices support applications requiring high-quality video over all digital networks and offer the options needed to develop high-end future applications. All these advantages lead to a reduced

system cost and ease-of-development.

The DM64x generation of digital media processors includes:

- Integrated high-definition capable video ports;
- Processing power up to four simultaneous MPEG-2 main-profile-at-main-level video decodes at full D1 resolution and 30 frames-per-second;
- Full MPEG-2 main-profile-at-main-level video encoding in real time;
- Industry's first code support for broadcast-quality Windows Media 9 encode;
- Support for the latest industry standard algorithms including MPEG-4 AVC (H.264) encode and decode;
- And other features such as Ethernet capabilities, multi-channel audio and 66-MHz PCI connectivity.

Nexperia- Philips

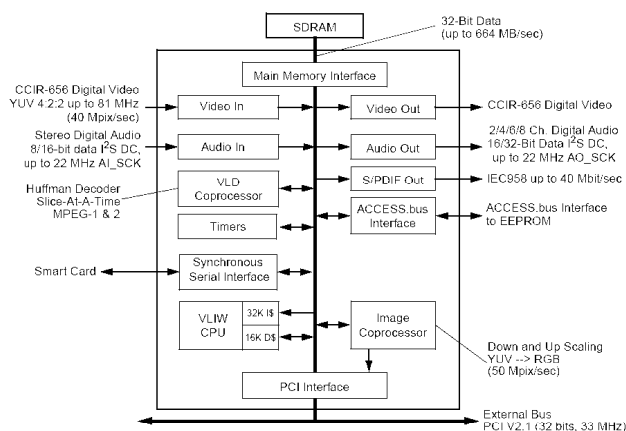


Figure 1. Nexperia PNX1300 structure

PNX 1300 is a programmable circuit with an architecture well optimized for applications requiring simultaneous processing of all data types from multimedia data streams.

With a large processing power for capturing, compress or decompress audio/video data in real-time, PNX 1300 is adapted for a broad spectrum of applications based on video processing: video conference, video editing, video database, control, industrial visual inspection systems and multifunction systems like digital TV and video surveillance systems.

Features:

- Data, audio, video, graphics and data streams are processed in a single circuit;
- Operating frequency is 143 or 166MHz, processing power up to 6,5 BOPS;
- Multiple instructions: traditional microprocessor, multimedia SIMD, IEEE floating point;
- Programming entirely in C, C++;
- Elementary processing libraries (video, images) delivered by Philips.

The complexity of Nexperia processors is suggested in figure1 (source: [2]).

2. FPGA

Field Programmable Gate Arrays (shortly FPGA) are high performance re-configurable components originated in digital IC domain not in programmable processor area. The high degree of flexibility and reconfiguration makes possible to reach the same performance compared with DSP areas, and in some cases even a better one. The FPGAs are used more like a co-processing structure attached to a main processor, but the latest generations of FPGAs contain processor cores, making possible to solve the entire application in a same chip.

The Embedded IP surveillance system that benefits of FPGA performance has roughly the following architecture (Figure2 [9]).

II. FACE RECOGNITION ALGORITHMS

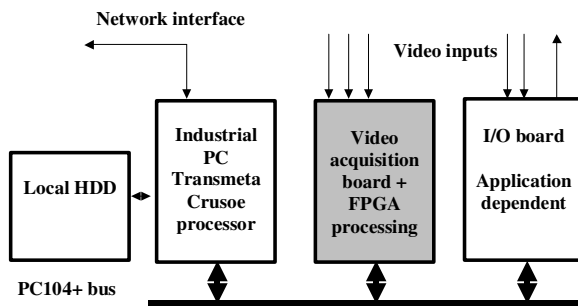


Figure 2. FPGA based surveillance system architecture [9]

The goal of face recognition is to determine the identity of an individual based on a still image or video sequence of his or her face. There are two different modes of operation for a face recognition system: authentication and identification. In the authentication mode, the system accepts or rejects the claimed identity of the individual. In the identification mode, the system compares the face image to a database of known people, and returns the most likely identity or identities.

Face recognition could be integrated in a wider perspective of image recognition methods [3]. In this

approach a typical recognition tasks has three stages (figure 3).

- Primary image transformation in a internal representation (could be interpreted as a preprocessing task or a mathematical transformation, for example eigenvalues calculation);
- Key features extraction (keeping only first n components for DCT, for example);
- Classifier (modeling): cluster based models, neural networks, etc.

In this manner the construction of recognition method is based on a-priori information regarding the image investigated (in this case human face) and corrected with experimental data, collected during investigation.

Based on whether the input is a still image or a video sequence, face recognition takes different approaches, each of which has its advantages and challenges. Figure 4 shows the block diagrams of two possible approaches to face recognition. For simplicity, the block diagrams from figure 4 assume that there is a single face in the given image or video sequence. In case multiple faces exist, the system should work

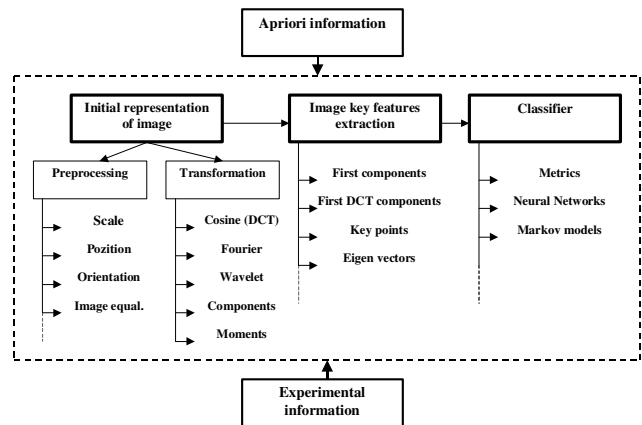


Figure 3. Methods for image recognition

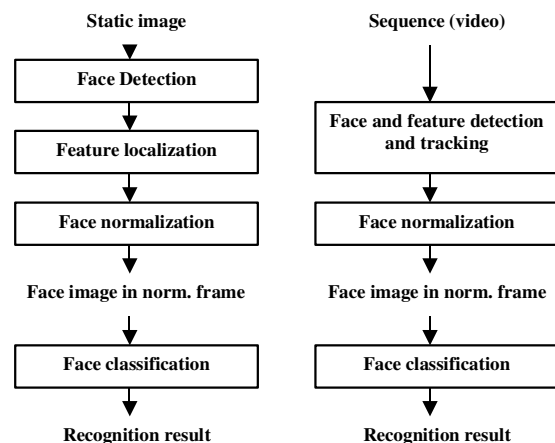


Figure 4. Two possible approaches of face recognition algorithms

on each of them separately.

With a still image input, the system whose block diagram is shown in part (a) of Figure 4 first finds the location of the face with a face detection module. Then, it searches for specific facial features, usually the eyes, to register the face image. Finally, the registered image is normalized, and a classification algorithm determines the identity of the person. Note that searching for the face and the features in still images is a computationally intensive task.

For face detection it is used the probabilistic visual learning approach proposed by Moghaddam and Pentland [1]. According to their approach, face images are modeled as a multidimensional Gaussian distribution that is estimated with the help of a Karhunen Loeve Transform (KLT) based dimensionality reduction. To detect faces in still images, blocks at different scales and locations are extracted from the image, and their probabilities of being a face are calculated using the density mentioned above. Since searching a large image at multiple scales and locations is a computationally intensive task, we tried to decrease the search space with the help of a rule proposed in [6]. Assuming that there is a single face in a given image, it is assumed that abrupt changes in the horizontal and vertical profiles of the image correspond to head boundaries. The horizontal profile is obtained by averaging the pixels at each column, and the vertical profile is obtained by averaging the pixels at each row. However, in the general case, if the person is in front of an arbitrary background, it is rarely possible to find the head boundaries with this approach because there are other abrupt changes caused mainly by the background. But, even in this case, the rule it is still useful to decrease the search space for the face. In the horizontal profile, the first and last abrupt changes are searched and it is assumed that at least half of the face is located between these two boundaries. If there is false detection due to the background, this just increases the search space without causing the miss of the face. Similarly, in the vertical profile, the first abrupt change is found, and assumed that the face is located below that upper boundary. The search process in the space between these boundaries at multiple scales and locations using the method proposed by Moghaddam and Pentland have explained above.

After the location of the face is found, eye localization is performed, where we search for the two eyes inside the face at multiple scales and locations. Again it is used the density estimation technique proposed by Moghaddam and Pentland, this time to model the distribution of the eyes.

After we find the eyes, we rotate the face image to make

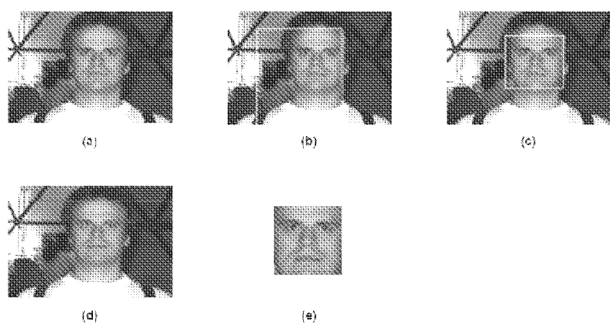


Figure 5. Face detection, eye localization and face normalization stages (a) original image, (b) search space reduction, (c) the result of face searching, (d) eye searching result, (e) normalized face image

the eyes horizontal, crop it to exclude the background, and decimate it down to a size of 128x128. These steps of face normalization are seen in figure 5.

After the face image is normalized, it is sent to a face classification algorithm that compares it to a database of known people and returns the most likely person. Two different face classification algorithms were implemented. The first one is the well-known Eigenfaces algorithm proposed by Turk and Pentland [4], which is considered to be a baseline algorithm for face recognition. According to this approach, face images are first projected into a subspace that is obtained by performing principal component analysis on the training images. Then, recognition is performed by minimum distance classification. The second classification algorithm we implemented is the segmented linear subspaces algorithm proposed by Batur and Hayes [5]. This algorithm's primary goal is to perform reliable face recognition under varying illumination conditions. According to this approach, each person's face images with a fixed pose under varying illumination are modeled with a segmented linear subspace model, and recognition is performed by computing the distance of the image to the subspace models in the database.

Of course, this algorithm or any other algorithm should be simple enough and use a small amount of memory to be suitable for implementation in an embedded system.

The proposed method was implemented (implementation described in [6]) for the enrollment and recognition procedures in Matlab and tested in the system on a subset of Database that contains a total of 300 frontal images of 10 people where the lighting direction changes between 0 and 50 degrees. For the automatic system the recognition rate for "Eigenface" method was 88%, and with the segmented linear subspaces classification was 93%. The two methods were implemented in a system based on TMS320C6416.

III. FACE RECOGNITION SYSTEM IMPLEMENTATION

Generic system (hardware) architecture

This section presents three possible generic architectures for systems applied in face recognition, described in [6], [7], [9]. The first two systems are based on media processors described in the first paragraph. The last one is a general image processing and pattern recognition system based on FPGA coprocessor architecture.

The presentation is focused on the possibility of implementation for face recognition algorithms in embedded systems, taking as examples databases with a limited number of faces, and not on very complex algorithms. The results are given for information, and not for the relevance of recognition algorithms or face databases.

The projected architecture is close to the reference architectures (for example presented in [6]) still based on a standard TMS320C6416 DSK board, with no special provisions in hardware architecture for this specific application.

Face recognition systems could be based on general-purpose platforms, like PC compatible architectures, or on stand-alone systems minimizing consumption, dimensions and costs. The trend is to use stand-alone systems, allowing placing and mounting in the most important places for an efficient identity control (airports, railway stations, access gates).

A generic system [7] of biometrics control system, is

based, as seen in the previous paragraphs, on a high performance processor.

Basic technical data of the generic (designed to realize a large class of image processing tasks) face recognition system are presented briefly in table 1. The flexibility of the DSP system allows, on the same hardware platform, to solve various applications in video processing domain: biometrics system, industrial vision, video surveillance.

Table 1

Image Resolution	Min. 640x480 (768x576) pixels
Input signal	Analog – PAL/SECAM compatible Digital – LVDS signals
Communication	Ethernet 100 MBit/s
Memory (Flash)	4-8 MB
Memory (RAM)	16-64 MB
Aux in/out.	8inputs , 8 outputs
Special functions	Direct signal visualization (compressed /uncompressed)
Options	Additional digital interface (USB, IEEE1394)

The main blocks of the recognition system have the following functions:

Conditioning block – perform analog signal multiplexing, amplifications, amplification, synchronization and black level restoration;

Acquisition and monitoring – is performing A/D conversion, digital video signal formatting in standards accepted by DSP video interface, imagine normalization;

DSP based CPU – has the overall control of the acquisition and recognition functions included in the application, communication control, threshold detection and decision, storage / restoration of information.

Software architecture

Software support is based on two main modules: OS core (real time OS preferably) and the main application.

OS core is imposed by the processor choice in implementation. For Texas DSPs the structure is based on DSP BIOS [1], common for DSP based applications in C6000 generation.

Philips Nexperia software development is based on TSSA (Trimedia Streaming Software Architecture [2]). In this architecture the recognition application is not a major part of the implementation. The support offered by TSSA allows having access to optimized image processing libraries, and a transparent method for data interchange in the system. A typical image of an application in Nexperia based systems is presented in figure 6. Main software blocks to power a face

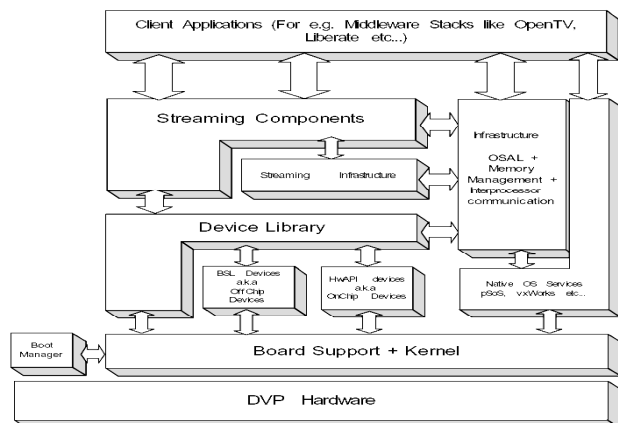


Figure6. TSSA architecture([2])

recognition system (server side) are as follows:

- OS core, Acquisition drivers
- Image normalization drivers
- Face recognition algorithms
- Classification
- Alarm handling, Communications drivers

Client module structure

The program used in main surveillance tasks, has few components driving the following functions:

- Positioning, optical zoom
- Alarm handling
- Reference / archive image download

FPGA based architectures

FPGA based architecture is a non-standard architecture, performing the face recognition task (or at least some of the processing phases) using the micro-programmed algorithm stored in FPGA. The architecture of the FPGA based board is presented in figure 7 (according to [7]). This is the dashed part of the recognition system described in block schematic from figure 2.

The processing phases are re-configurable, the main processor (in this case a PC104 compatible CPU) being able to modify, even dynamically, the processing flow through FPGA.

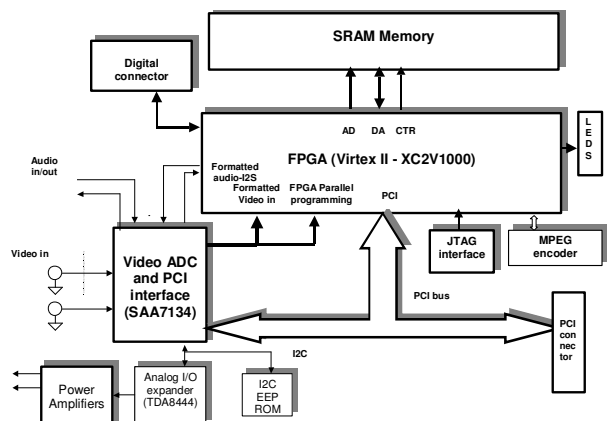


Figure7. FPGA based processing architecture [7]

IV. PERFORMANCE AND EVALUATION

The main issue in this evaluation is to verify if a simple, embedded system could be used to implement face recognition systems. The increasing number of such systems described in the literature gives the answer.

The application presented in [6] was tested on a sample image of 640x480 pixels containing a single face. The database is composed from 10 faces as presented. Taking into account the tiny dimensions of the database the search process is performed in main memory.

Since the rule based approach used for decreasing the search space for face detection causes variability in computation time, the performance results are averaged over a certain number of input images. The resulting CPU and memory requirements are shown in Table 2. These results revealed that the face detection and eye localization blocks consume most of the computation time, and the face classification blocks consume most of the memory. These results are expected since searching for faces and features in

still images at multiple scales and locations is known to be a computationally intensive task, and the classification blocks have to store the subspace models and the face databases which are large in size.

A brief look to the results shows that it takes less than 4 seconds to recognize a face. Most of this time is spent during the face detection and eye localization stages. Of course, choosing faster algorithms for these stages could increase the recognition speed significantly.

Table 2- Results of the implementation

Stage	Cycles(x10 ⁶)	Time (CPU 500MHz)	Memory usage
Face detection	1161	2.32 s	~ 392 KB
Eye localization	585	1.17 s	~ 436 KB
Face normalization	56	0.11 s	~ 32 KB
Face classific. (Eigenfaces)	18	0.04 s	~ 1055 KB
Face classific. (Segmented linear sub.)	22	0.05 s	~ 2064 KB

Regarding the method efficiency, Eigenfaces classification is faster, consumes less memory, but is providing a lower recognition rate than the segmented linear subspaces method. Similar tradeoffs would probably exist for all face classification algorithms.

V.CONCLUSION

Face recognition task is one of the most complex problems from image recognition field. It implies face anatomy knowledge and requires complex preprocessing and classifying algorithms. Embedded face recognition systems are still at the beginning. Rapid expansion is difficult, especially analyzing the major difficulties still existing in processing algorithms and the unsatisfactory recognition rate. In these cases storage or transmission of dubious images for "off-line"

analysis (eventually assisted by a human operator), could be an option for false recognition rate improvement.

Increasing processing power of high performance DSP (Media Processors) or FPGAs, allows implementation in an efficient way, from price /performance ratio, dimensions or power consumption. Improvement of the processing algorithms will allow for the embedded systems to be a real, effective alternative to well-known surveillance cameras.

REFERENCES

- [1] * * *, "TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor, *Data Manual*, Texas Instruments", 2002
- [2] * * *, "PNX1300 Series Media Processors", *Preliminary Specification*, Philips Semiconductor, 2002
- [3] Д. Брилюк, В. Старовойтов, „Методы распознавания человека по изображению лица. Достоинства и недостатки, сравнение”, *website: <http://daily.sec.ru>*, 2002
- [4] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1991, pp. 586-591.
- [5] A. U. Batur and M. H. Hayes. "Linear Subspaces for Illumination-Robust Face Recognition", in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, pp.296-301.
- [6] Aziz U. B. and B. E. Flinchbaugh, „Performance Analysis of Face Recognition Algorithms on TMS320C64x”, *SPRA874, Texas Instruments*, dec. 2002
- [7] R. Arsinte, "Face recognition module", *Technical report* , Microtech, 2003
- [8] R Arsinte, "Implementarea sistemelor biometrice autonome pentru recunoaşterea feţei, utilizând procesoare DSP media", in *Proceedings of the Workshop Verificatori Biometrici, Cluj-Napoca* , 2005, pag.135-143 {in Romanian}
- [9] R.Arsinte, Implementing a Test Strategy for An Advanced Video Acquisition and Processing Architecture, in *Acta Technica Napocensis – Electronics and Communications*, nr.2/2005, pp.15-18