SYSTEM LEVEL DESIGN OF A PATTERN RECOGNITION SYSTEM BASED ON THE GABOR WAVELETS

Mourad Aberbour, Habib Mehrez, François Durbin*, Thierry Garié*, André Tissot*

Laboratoire LIP6/ASIM UPMC Paris VI, 4 Place Jussieu, 75252 Paris Cedex 05
*CEA-DAM, SEIM, Centre d'Etudes de Bruyères-le-Chatel, BP 12, 91680 Bruyères-le-Chatel
e-mail : mourad.aberbour@lip6.fr

ABSTRACT

In this paper we will address the system level design of pattern recognition architectures. The system targeted uses Gabor wavelets for the multi-frequency and multi-scale analysis of gray scale images. The macro-blocks or macro-functions architectures are developed in sight of the integration of a system-on-a-chip.
We have faced two basic design problems. First, to which extent the algorithms can be simplified to fit a hardware implementation? Second, how the functionality will be partitioned to fit an optimal hardware/software system?
To carry out these operations we have relied on a design methodology taking into account the realization constraints early in the design stages. And we used a fast cycle precise simulator to attempt and evaluate many hardware/software alternatives.

1. INTRODUCTION

The aim of this work is to study and design an embedded recognition system. The application targeted is face recognition in the beginning. The algorithms with slight variations have been used for other applications like texture classification. We have started from a specification of a pattern recognition system relying on the use of the Gabor filtering and the extraction of the pertinent information using saliency measures. A simple perceptron was used for learning and classification. To enhance the performances of the system we use a Radial Basis Functions (RBF) neural network.
Artificial vision systems must take into account several types of irregularities and deformations that may affect the object to be analyzed. These variations are orientation, scale, translation, perspective, lighting and occlusion. It is clear that a biological system is invariant to these factors.
In 1946, Gabor presented a new way to analyze temporal non-periodic signals [1]. His idea is based on the representation of the signal characteristics in a time-frequency space. He then introduced the functions known now after his name, the Gabor functions. In the 80s, the ideas of Gabor have been exploited by Morlet and Daugman and applied to two dimensional signals[2]. Some studies on the mammal vision process showed that such a system behaved like a two dimensional filter with an impulse response close to the Gabor functions. The paper of Jones, Larry and Palmer[3] presents in much detail the results which generalize this approach and discuss its limits. In this paper we will first present some of the theory behind the Gabor wavelets and then layout the algorithms used. After that we will focus on the design of the system. We discuss the algorithm/architecture adequation and system-level design issues.

2. THE GABOR FILTERING

In our system the information extraction relies on the Gabor filtering. The corresponding filters are constructed from dilatations and rotations of a Gabor type mother wavelet.
Let's remind the definition of the impulse response of a Gabor filter

$$G(x, y, \sigma, k_x, k_y) = e^{\frac{-\left(k_x^2+k_y^2\right)}{2\sigma^2}\left(x^2+y^2\right)} \bullet e^{j\left(k_x x+k_y y\right)}$$

where σ = spatial resolution of the filter
and k = spatial frequency.
The Gabor functions as defined originally keep a constant spatial extent (for a certain Gaussian envelope) and the number of waves under this envelope varies. It is nevertheless interesting to define time-scale wavelets derived from a mother wavelet by dilatation and translation. Transposing these properties on the Gabor wavelets implies a constant ratio between the wavelength and the spatial extent of the modulating window, the Gaussian. For this reason Morlet has defined the k and σ as well the scaling factor N and upgraded the Gabor mono-dimensional functions to the two-dimensional wavelets.
We then obtain the Gabor-Morlet wavelets satisfying the time-scale conditions. These are defined by the following equation:

$$G_{Morlet} = \frac{1}{N_{k,\sigma}} \bullet G(x, y, \sigma, k_x, k_y)$$

For the extraction of the information in our images we use the Gabor filtering. We distinguish this from the decomposition of the signal under study into a basis of Gabor functions which we would call the Gabor expansion. The filtering must permit the extraction of the object's characteristics.

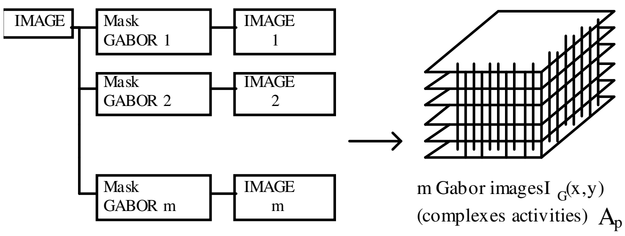

Figure 1. Computation of the Gabor Activities

The image is filtered with a bank of filters at different spatial frequencies (k) and at specific directions (θ). The result of this filtering at a point is called a **Gabor activity**.
From each point on the original image we extract a corresponding activity. These activities will build up a representative vector of the point and its immediate

neighborhood. It is necessary to precisely choose the filters characteristics (spatial frequency and orientation). Getting the m Gabor images, is equivalent to getting a vector of m components for each pixel of the initial MxN image. Thus we must store MxN vectors, each of which is of m components (fig. 1). These MxN vectors are too many to be easily exploitable. This is why it is necessary, at this stage of analysis, to extract those that best characterize the initial image.

3. IMAGE CHARACTERISTIC ACTIVITIES EXTRACTION METHODS

There exist in the literature several methods for the extraction of pertinent information in an image [4,6]. We will present here the chosen method that is the Clark & al method [4]. It is based on the saliency notion. It has been chosen for its adequacy for a hardware implementation (this constitutes the first algorithmic choice guided by the architecture) in regard of the data precision and relative regularity of the data processing in general.
The saliency is defined to be a measure of the specificity of the information in a vector of Gabor activities.
We define the normalized and the non-normalized saliencies of two vectors V_i, V_j by these two formulae:

$$S(V_i, V_j) = \frac{V_i \bullet V_j}{\|V_i\| \|V_j\|}$$

for the normalized saliency corresponding to the cosine of the angle between the two vectors. And

$$S_u(V_i, V_j) = V_i \bullet V_j$$

for the non-normalized saliency.
The saliency of a vector (normalized or not) is defined as:

$$S(V_i) = \sum_{V_i \in 3x3} S(V_i, V_i)$$

for a certain neighborhood.
This quantity is defined for all the vectors at all points of the image. The vectors are then ranked in the descending order according to their non-normalized saliencies. This list must be shortened considerably. Eliminating the vectors of saliencies smaller than a certain threshold will eliminate all the points corresponding to the background of the image which contains no significant information relative to the object of interest. However the pixels close to the object require further processing. The spatial extent of the Gabor filters is inversely proportional to the spatial frequency. Thus, for the low frequencies, we filter the images with bigger masks that generate Gabor activities of big values. And the corresponding jets cannot be removed by the previous processing.
We define a ratio R between the first m1 components of a characteristic vector corresponding to the higher spatial frequencies (low extent), and the m1 components corresponding to the low spatial frequencies (R Criterium). If this ratio is inferior to a certain threshold then we can suppose that the big value of the saliency of that vector is due to the spatial extent of the filters rather than that the vector belongs to the object of interest in the image. So we do not keep this vector. In the opposite case, the pixel under study belongs to the object and we keep the associated value. The choice of the threshold defines the number of eliminated vectors, and their proximity to the object.
The above selection is still insufficient and must be enhanced. We now seek to keep a vector and only one in a certain neighborhood (3x3 for example), we call this paving. The vector of largest saliency will be kept. We apply this method to all vectors remaining after the R ratio criteria. Once this operation finished we are left with only the pixels belonging to the object under analysis. We must then determine among the remaining vectors the ones that are most significant.

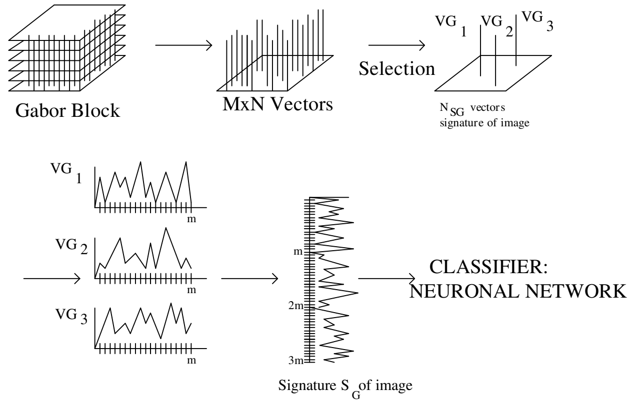

Figure 2. Clark & al. Saliency Algorithm

First we form clusters of vectors according to their normalized saliencies which evaluates the parallelism of the vectors. Second we keep for each cluster the vector having the largest non-normalized saliency. This last set when concatenated will build up the signature of the image, figure 2.

4. ALGORITHM/ARCHITECTURE ADEQUATION

From a theoretical specification of the algorithm we have studied its hardware implementation. The challenge was then to transform or readapt the algorithm to make it adequate for an optimal hardware structure without affecting its performance and capacity to resolve the problem.
The recognition system has been modeled completely at two different abstraction levels, a functional and an architectural level. Functional simulators have been realized. These are divided into two categories. Those implementing the application as is without any modification, and those which take into account the enhancements or compromises guided by the architecture. The functional simulators are written in standard C and use the standard data types (int, float, double, …). The reference model is then the first simulator which simulates the complete system according to the original specification of the algorithms and complying to the mathematical rigor.
The architecture simulators model the hardware structure. They are inspired from the architecture. Their performance is compared to the functional simulators. Functions controlling the data widths and well tailored types are introduced.
With this set of simulators we achieve a complete parametrization of the system ensuring the best algorithm/architecture compromise. The choice of the different parameters is defined by the sought performances of the system. A set of tests is carried out to ensure the consistency of the architectural and functional models. We distinguish two categories of parameters to monitor: execution parameters and data dynamics parameters. These are discussed in the next sections.

4.1 Execution Parameters Control

The first execution parameter consists of the number of filters used in the analysis of the images. This defines the number of orientations. The number of angles to span depends on the objects targeted by the application.
For very textured objects we will need a very tiny analysis at the orientations levels. For objects with coarse textures and very regular details, a limited number of orientations would be sufficient. For face recognition we would at least use six orientations and 4 resolutions. A research group at the university of California use 40 filters with 8 orientations and 5 resolutions[5]. The Gabor filters are complex. For a mathematical rigor we should use the real and imaginary parts

in the analysis. But ignoring the phase would simplify and alleviate the computation load of the system.
In the paper of Clark et al. the complex filtering is computed by this formula:

$$FG\left(i,j,k,\sigma\right)=Re\left(i,j,k,\sigma\right)+i\cdot\operatorname{Im}\left(i,j,k,\sigma\right)$$

and each jet is build up using the magnitude of this complex quantity:

$$\left\|FG\left(i,j,k,\sigma\right)\right\|=\sqrt{Re^2+\operatorname{Im}^2}.$$

In a first time, and to simplify an architecture intrinsically complex, we have adopted the following approximation:

$$\left\|FG\left(i,j,k,\sigma\right)\right\|\cong\left|Re(i,j,k,\sigma)\right|$$

justified by the first tests concerning face recognition. This is also discussed in [5].
The number of vectors eliminated at each phase of processing depends on certain values fixed after extensive tests on a certain image reference library. Finally the number of vectors which will constitute the signature must be fixed carefully to ensure best performances of the system. All these parameters are fixed after extensive simulations with a certain images database.
The final algorithm for the extraction of the final signature is an optimization problem. The computations necessary to carry out this operation are very complex. Their implementation would penalize all the previous optimization efforts. It has been necessary to simplify this algorithm. We finally use the selection of the final vectors according to their non-normalized saliency. Taking into account that the paving and R criterium algorithms ensure us a disparity of the subsisting vectors.

4.2 Control of the Data Dynamics

The data dynamics processed by the system influence directly the complexity of the architecture. To ensure an optimal architecture keeping reasonable performances, it is necessary to judiciously fix the precision of the data processed. Since the images are coded on 8 bits and the filters on 13 bits, and knowing that we can have 32x32 filters, the a priori precision of the convolution results would be 32 bits. But after studying the Gabor filters we could show that the results would not exceed 22 bits in precision. Moreover these results on 22 bits can be further normalized on 16 bits. Using the architectural simulators we could show that this optimization is possible. This reduction allows a considerable gain in the architectures for the selection of the pertinent vectors and in the architecture of the neural network.
Many enhancements and optimizations have been possible by the extensive use of both the functional and architectural simulators. The reference performance was the one achieved by the functional and fully theory complying simulators.

5. The Hardware/Software Codesign

We are faced in this project with the problem of the design of a very complex system-on-a-chip. From the specific knowledge we acquire on the targeted application we can design a system to realize it and that is optimized in performance and more efficient than a general purpose system.
The implementation of a specific application begins with a specification phase. The designer establishes the computing structures and the coarse modules which permit a clearer and simplified view of the application. These modules are usually operative blocks (data-paths) realizing a certain function. The complete system will be the association of these different functions, also called CoProcessors, with communications among them. Today's systems are a combination of hardware and software components. Every nowadays designer is faced with such issues as how to partition the application into hardware and software. How to integrate the macro-blocks (known also as IP blocks) and how to interface them together to achieve the highest performance system. All these issues consist the operations of system level design.
The partitioning has been a trial and test procedure. No automatic partitioning tools exists nowadays. Recently Professor Hugo De Man stated "Being an academic, I know there is a lot of work being done on automatic partitioning, but I don't believe in it".

6. Global Prototype of The System

For the design of our system we have relied on the results of COSY, a European ESPRIT project. COSY aims to provide a design methodology and platform for the simulation, synthesis, and performance evaluation of RISC/DSP embedded systems that merge control and signal-processing functions into a single chip. A tool for cycle precise simulation of a system based around the PI-bus and a set of generic interfaces have been developed [9,10]. A prototype system is shown in figure 3.

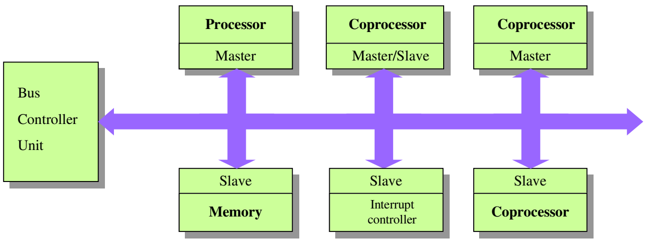

Figure 3. Typical PI-bus system

The PI-bus is a multi-master on-chip bus. This standard is developed jointly by many European leader companies. It furnishes a standardized communication interface for communication between the macro-blocks [9]. We test the system using the same benches of the functional validation phase.
The computation load of our system is of the order of 50 GOPS. The data communications in such a system are not negligible. We use FIFOs to bufferize and get the data on-chip to minimize the data communications and the very penalizing memory accesses. In the near future the memory accesses will be a big problem. This is due to the fact that the access time to memory is lagging more and more the computation speeds of the operative parts of a system.
One of the complex problems with the design of vision systems is the variation of the data structures processed over the system. In our application for example we start by a two-dimensional data structure, the images. And then we build a 3D structure, the Gabor block. Then the system will start processing vectorial structures. Finally the neural network must process a set of vectors with a certain representation of its internal data.
Once an optimum hardware/software architecture is obtained it must be validated before going to the physical realization. The first validation consists of writing a model of the architecture. The model is written in a high level language that implements the architectural aspects using procedural operators and which describes the cycle-precise behavior of the system.
The architecture is data-flow oriented. Thus the synchronization of the parallel processing is implicit. The communications between the different functions are simplified.
The recognition method under study necessitates the convolution of an input image with a certain number of Gabor filters. A VLSI ASIC convolver has been designed and fabricated for our application [11]. This convolver is also available as a macro-block for instanciation as a module in a larger system, and was fully designed under the GenOptim environment [12], which allows the design of portable macro-blocks generators. The architecture of the convolver used some

flexible and performant concepts: use of systolic arrays, vectorization of data, cascadability and finally reconfigurability. The convolved images must be memorized for subsequent processing. The whole system is presented in figure 4. Beyond the study and realization of the different coprocessors, a hard challenge imposed to us was the definition of all the communications. In fact in this system most of the processing is carried out by different components each having a very precise functionality.

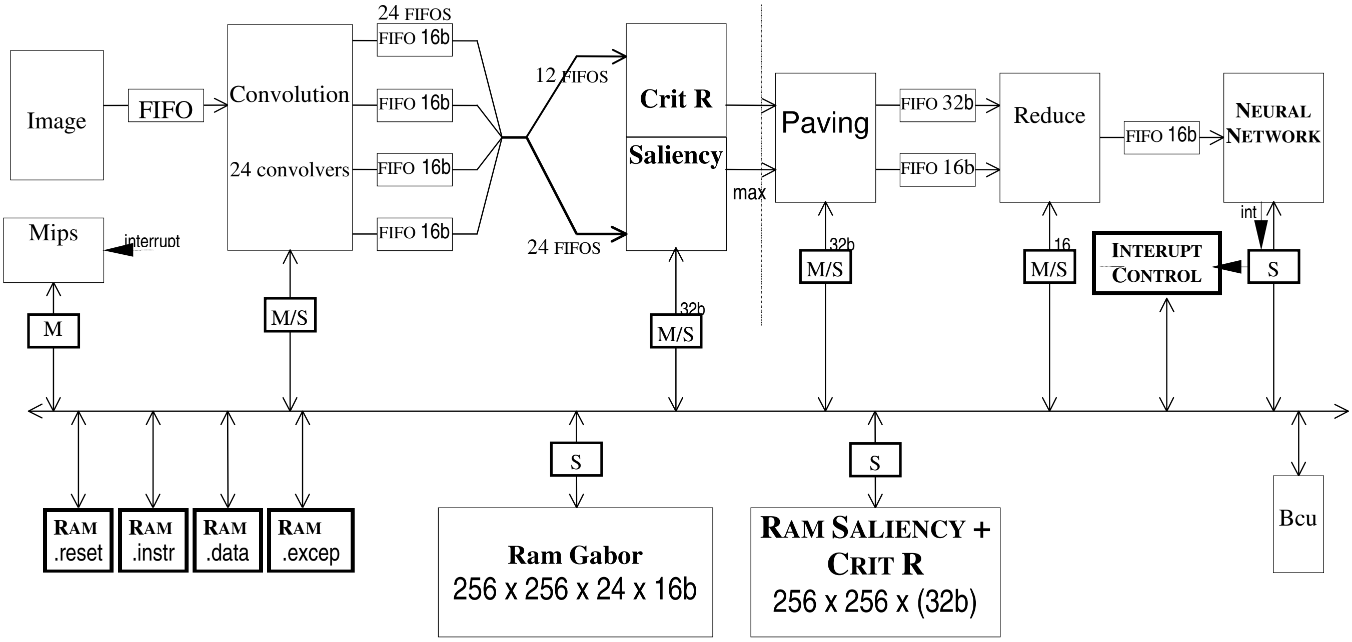

Figure 4. Global System Architecture

All the communications is achieved through FIFO interfaces. Each coprocessor has an interface to the bus. A Master interface for the MIPS, a slave interface for the memory and Neural Network, and a Master/Slave interface for the convolution, saliency, paving,…The MIPS has been kept to carry out a minimal amount of computations. It is used as a manager of the whole system. It issues orders to the coprocessor and processes the different exceptions and interruptions. Many software routines are executed by this processor, for example the learning algorithm of the neural network. Since the saliency and the R criterium coprocessors use the same data, and they must write the same data relative to the same pixel at the output, they have been tied together in the same component. Instead of writing the result of the R criterium in memory, it is used directly to invalidate the pixel with disregard of its saliency. The paving and reduction coprocessors output a reduced number of vector positions to be fetched from the Gabor memory and fed to the neural network for learning or classification.

7. CONCLUSION

We have addressed in this paper the problem of designing a complex system-on-a-chip for pattern recognition. The system uses the Gabor filtering for analysis and saliency measures for signature extraction. We use in this system an RBF neural network for learning and classification. We have tailored the algorithm to fit a realistic and perfomant implementation. The paper has been focused on the algorithm/architecture adequation and hardware/software codesign issues. The extensive use of functional and architectural simulators has shown to be of great value.

The system, with the convolution macro-blocks designed for the Gabor filtering and the RBF neural network, can serve, due to its programmability to implement a variety of other types of recognition systems based on the Gabor analysis and neural networks.

8. REFERENCES

[1] Gabor, D., *Theory of Communication*, Journal of the IEE 93, Part III, no. 26, pp. 429-457.

[2] Daugman J.G., *Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression*, IEEE ASSP, vol. 10, No. 2, 1988, pp 1169-1179.

[3] Jones, Larry, Palmer, *An Evaluation of the Two Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex*, Jour. Neurophysiology, vol. 58, No. 6, 1987.

[4] Clark, G., & al, *Gabor Transforms and Neural Networks for Automatic Target Recognition*, Workshop on Neural Networks, 1991.

[5] Von Der Malsburg et al., Distorision Invariant Object Recognition in the Dynamic Link Architecture, IEEE Transactions on Computers, vol. 42, No. 3, March 1993.

[6] Flaton, K. A., & al, *An Approach to Image Recognition Using Sparse Filter Graphs*, IJCNN 1, pp 1313-1320, 1989.

[7] Child, J., *Image Processing Architectures Take New Twists*, Computer Design Magazine, March 1997.

[8] Ruetz, A., & al, *Architectures and Design Techniques for Real time Image Processing IC's*, IEEE Journal of Solid State Circuits, vol sc-22, No.2, 1987.

[9] Pétrot, F., Hommais D., Greiner A., Cycle Precise Core Based Hardware/Software System Simulation with Predictable Event Propagation, Proceedings of the 23rdEuromicro Conference

[10] Open Microprocessor Systems Initiative, Draft Standard, OMI 324: PI-Bus, Rev. 0.3d

[11] Aberbour, M., Houelle A., Mehrez, H., Vaucher, N., *A Real-Time Image Convolver,* International Conference on Signal Processing Applications and Technology, Boston, Oct. 1996.

[12] Houelle, A., *GenOptim: un environnement d'aide à la conception de générateurs de circuits portables optimisés en performance et en surface*, Ph.D. Thesis Paris VI, 1997.