REAL TIME MOTION DETECTION ON DYNAMIC SCENES STUDY AND IMPLEMENTATION WITH RECONFIGURABLE TECHNOLOGY

M. MOSTEFAI[1], B. ZITOUNI[1], M. ROUSSEL[2]

[1)] Institut d'Informatique, Université Ferhat ABBAS - Setif - Algérie -

Tél/fax : 0 35 68 28 45 - Email : espace@mail.wissal.dz

[2)] Université de Reims (France)

ABSTRACT

Moving objects extraction from dynamic scenes is a well used approach in applications that requires high compression rates such as surveillance. Unfortunately, the different operators developed to undertake this operation and that use often classical edge detection techniques in their processing remains very sensitive to noise. Different approaches where used to resolve this problem. Among these, fuzzy approach is recently used to develop robust operators less sensitive to noise.

The aim of this work is to present a fuzzy edges detector used by a moving objects extraction operator, and its implementation within an FPGA circuit of XILINX for a real time images processing.

Key words : Image processing - Motion detection - Fuzzy logic - Real time - Reconfigurable technology

RESUME

L'extraction d'objets mobiles à partir de scènes dynamiques est une technique très utilisée dans les applications qui exigent des taux de compression élevés telle la surveillance. Malheureusement, les différents opérateurs développés pour effectuer cette opération et qui utilisent souvent des techniques d'extraction classiques sont très sensibles au bruit. Différentes approches ont été proposées pour résoudre ce problème. Parmi celles ci, l'approche floue. Cette dernière est récemment employée pour développer des opérateurs robustes moins sensibles au bruit.

Le but de ce travail est de présenter un détecteur de contours d'objets mobiles basé sur l'approche floue ainsi que son implémentation dans un circuit de type FPGA de XILINX pour un traitement d'images en temps réel.

1. INTRODUCTION

With the aim to satisfy the real time processing requirements, various image processing systems proceed to an important data reduction to compensate the insufficient power of current processors [1]. On the other hand, in many applications (tracking, surveillance, guidance...) alone the position and form of objets is mannered, consequently the alone knowledge of moving edges is sufficient. To proceed to a moving edges analysis, it is necessary to use an operator that, applied to an image sequence allows to extract from a dynamic scene the edges of objects in movement. For this purpose, different operators have been developed [2][3][4][5]. They use often in their processing classical edge detection techniques, easy to implement but sensitive to noise. The relative failure of the classical detectors has behaved us to choose another segmentation approach that use on of fuzzy tools to perform a fuzzy segmentation.

Henceforth, applied in many areas (process control, expert system ...), fuzzy logic brings equally its evidence in image processing. segmentation, image enhancement and edge detection [6] [7] [8].

We present here an approach which possesses the particularity to make cooperate contour and region approaches and permit to construct robust operators.

After a brief introduction to the moving edges detectors, we will present the basic principles of the fuzzy approach and its application in the construction of an edges detector. Software results will be presented as well as the implementation of this last within a XILINX FPGA circuit for a real time image processing.

2. MOVING EDGES DETECTION OPERATORS

Two categories of methods are used for the detection of moving objects in a sequence of images : Those that make a comparison of the current image with a reference image that represents the static model of the filmed scene [2][3].

Those that compare the current image with the neighbouring images, by concentrating directly on zones where changes are observed [4][5].

The construction of a reference image is an often difficult task, because variations of lighting within the analysed sequence can make some images incompatible with the reference image. On the other hand the comparison of neighbour images does not introduce relative constraints to the filmed scene and is less sensitive to variation of lighting.

Haynes and Jain [2] suggest an operator Ren(x,y) that perform the product of the the absolute difference of two succesive images (In-1 and In) with the gradient of the current image In. The expression of this operator is described by the next formula :

$$\forall x,y\ Rn(x,y) = \left| In(x,y)-In-1(x,y) \right| * Gn(x,y)$$

with Gn the gradient of image In. Rn : the result of the operation which correspond to the moving edges. This opertor does not performs a coincidence between contours but rather between the contour of the current image and the absolute value.

Another operator proposed by Stelmaszyk [3], consists in a multiplication between the gradient of the difference of two succesive images (In-1 and In) and the gradient of the current image In, allowing thus to have a true coincidence between contours. The formulation of this operator is as follow :

$$\forall x,y\ Rn(x,y) = G(\left| In(x,y)-In-1(x,y) \right|) * Gn(x,y)$$

In general, operators that operates only on two consecutive images induce false contours. Indeed, in this case, any visible object on the curent image which was not found on the one before, is considered as mobile, even if it concerns an element of the static bottom that was hiden, and that comes to appear (fig.1).

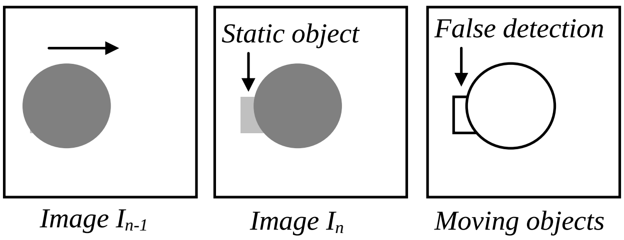

Figure 1. Occultation problem.

So as to avoid this problem, the processing must operate on three consecutive images. Vieren [4] has then proposed an operator which makes a product of the absolute differences of couples (In-1, In) and (In, In+1). The formulation of the operator is as follow :

$$\forall x,y Rn(x,y)=G(\left| In(x,y)-In-1(x,y) \right|)*G(\left| In+1(x,y)-n(x,y) \right|)$$

Another operator using three consecutive images in its processing was also proposed by Orkisz [5]. This last, allows mobile frontiers in the current image to be located by comparaison of frontiers of the current image to those of next and preceding images so as to distinguish mobile elements of static ones. The expression of this opeator is as follow :

$$\forall x,y Rn(x,y)=max(Gn(x,y)Gn-1(x,y)Gn+1(x,y))-max(Gn-1(x,y)Gn+1(x,y))$$

with *G*n, *G*n-1 and *G*n+1 respective gradients of three consecutive images In-1, In and In+1.

From all known operators that use varying elements between neighbour images, we have shosen Orkisz operator; this last represents the best compromise between efficiency and simplicity of implementation. Indeed, compared to other operators (eg: Vieren operator) that consist in performing products of gradients, the Orkisz operator requires only comparison and substraction operations, easy to implement and not affecting the dynamic as with multiplication operation.

As all the other operators this last use a classical edge detection technique (gradient), easy to implement but sensitive to noise. In add, this operator produce coefficients with a format size superior than 4 bits which oblige us to use large size memorys to store previous and current image

The objective of this work is to endow the chosen operator with a fuzzy edge detector, less sensitive to noise, and with reasonable implementation costs.

3. THE SEGMENTATION METHOD DESCRIPTION

3.1 Basic principle of Fuzzy logic

Fuzzy logic find its application when the information to process becomes uncertain [6]. This situation is often meet in image analysis, when the transition between two grey level zones is not clearly distinct. Edge extraction operation appears thus a difficult task. We can however emancipate to this by using the fuzzy reasoning which select edge point from a set of probable points. For this, we have to define data classes that we label (linguistic classes) and in which we arrange the information (linguistic variables) by giving a weight (membership degree) to each of these classes. The association of a linguistic variable *x* to a fuzzy class A can be thus formalised by the relation :

$$f_A : A \rightarrow [0,1]$$

$$x \rightarrow f_A(x)$$

Where $f_A(x)$ is the membership degree for the fuzzy class A. Once that the linguistic variables as well as their associated membership functions have been defined, a fuzzy operator will be used to proceed to an action according to its entries and by using pre-established rules.

3.2 Data partitioning

Consider an image In having gray levels in the interval [0,G-1]. It is assumed that each processed pixel $I(x,y)$ has eight closest neighbours (Ii, i=1,8). The partitioning process consists on computing for each central pixel I the difference luminance set : $D = \{d\text{i}\}$, with (di = Ii - I).

These values will be used as input variables by the fuzzy edges detector to compute the new central pixel value $I'(x,y)$. The variation domains of di and $I'(x,y)$ are respectively the closed intervals [-G+1, G-1] and [0 , G-1].

The input variable di can be member of several classes with membership degrees varying between 0 and 1. The membership functions can have several forms. However triangular and trapezoidal forms are the most used, allowing thus to have easy algorithms to implement. In our case three linear membership functions are used (figure 2). The data partitioning operation allows to allocate each input variable to one of the three classes (class0, class1 or class2) by using the function Max(μp,μg,μc). μp, μg and μc corresponds respectively to the membership functions of class 0, class1, and class2.

The di values belonging to the class2 are called dynamic values and will be arranged in one of the two static classes "0" or "1" only if their two close neighbours belongs to the same class "0" or "1"; otherwise, all configuration including these values is not taken in account in the phase of rules evaluation.

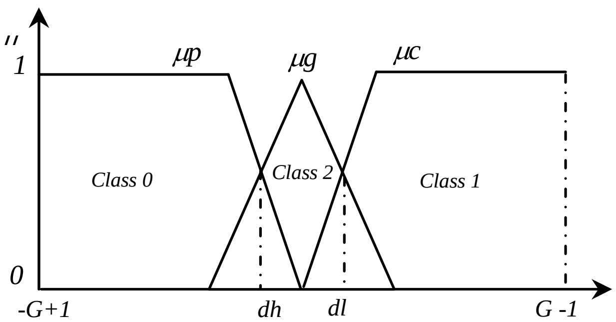

Figure 2 : Used membership functions

After the data partitioning operation the operator will use the previously generated fuzzy rules and the input variables allocations to compute the new central pixel value $I'(x,y)$ (equal to "1" if the central pixel is considered as a contour point and "0" if not).

3.3 Fuzzy rules generation

To elaborate our fuzzy rules, we have used the popular IF-THEN approach well described in [7]. This last use a group of N THEN rules (ending to the same result) and one ELSE rule (figure 3).

The fuzzy rules generation is undertaken from a 3x3 size binary mask. The eight examined contour types are presented in figure 4. An operator is used to examine the considered point neighbourhood. In the case or one of the eight configurations is met, the considered central point is taken for a contour point otherwise, it is considered as belonging to a uniform zone.

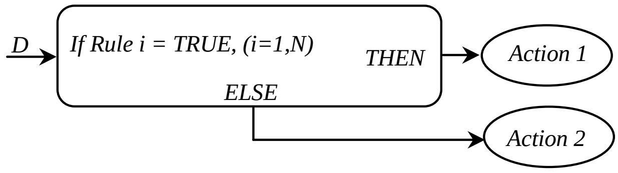

Figure 3 : IF-THEN approach

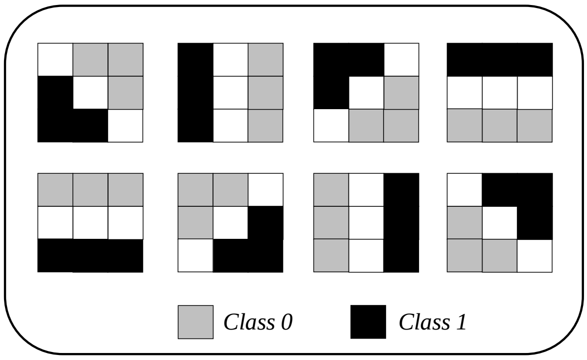

Figure 4 :. Examined contour types

4. SIMULATION RESULTS

In this study we are interested to real world scenes that have the reputation to be noisily an with weak dynamic. In order to see the performance of the fuzzy approach, we have applied the Orkisz operator to a real sequence of urban traffic and compared the result obtained by using a classical edge detector (Sobel operator) with that obtained by using a fuzzy edge detector (figure 5). As we can see it, the use of the fuzzy approach allows to have noiseless scenes with thinner edges. These performances are especially linked to the fact that this method uses a strong relationship between regions and contours. The use of the fuzzy approach allows effectively to improve results obtained by the classical approach, but remains also ineffective in some particular cases. Indeed, a same scene behaved to different results in natural light and in artificial light; shade of objects induce wrong contours, notably for objects in movement. A uniform lighting of the scene as well as a smoothing operation of acquired images are therefore necessary to obtain noiseless contours and more representative of objects in movement.

5. HARDWARE IMPLEMENTATION

5.1 Design Considerations

All design and implementation process has to take in account the constraints imposed by the chosen technology and the aimed application.

The design and implementation of real time image processing algorithms require making trade-offs among various design considerations.

the first design constraint is the integration chip capacity. In the framework of this study the reconfigurable technology is chosen. This last allows effectively to reduce the time and cost of the implementation, but remains again limited from the capacity of integration point of view. This limitation is more apparent especially when we have to deal with algorithms that use memory in their processing. Indeed FPGA RAM memory consume Clbs (reconfigurable blocks) resources that could other-wise be used to implement logic, and large RAM may restrict the amount of logic that can be included in the circuit [9].

In the case of applications that require great size RAMs, the use of external memory become necessary even if this solution increase the number of I/O pins needed on the FPGA, and decrease the application performances.

Another principle design constraint is the real time operation requirement. In general, a process is said to be operating in real time if it processes data at the same rate as it is presented. In our case the video signal is sampled at a rate of 1 pixel per 100 ns, giving 512 lines of 512 pixels per image, and 25 images per second. This sets the global clock rate for the processor at 10 MHz.

The real time constraint is largely supported by the XILINX FPGA families and more particularly the XC4000 families which achieve high speed through advanced semiconductor technology and through improved architecture, and supports clock rates of up to 50 MHz.

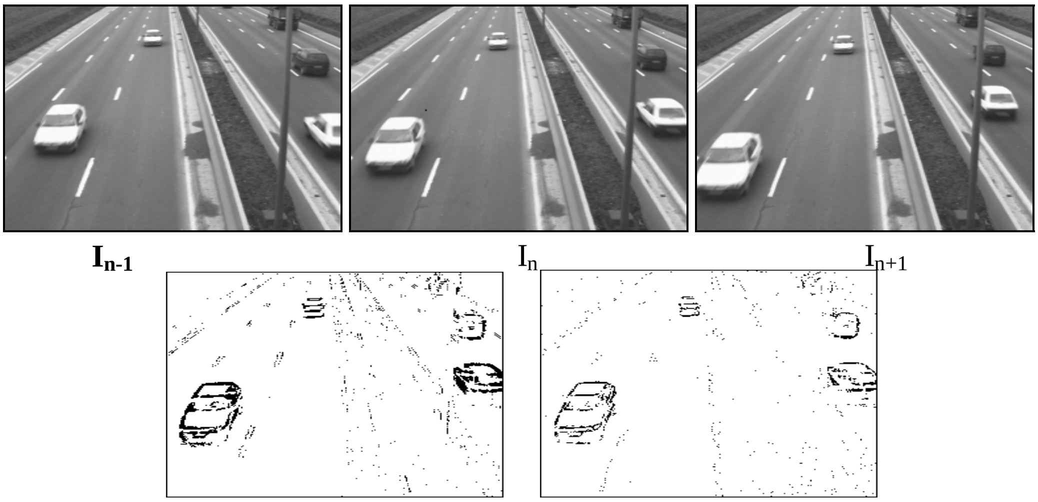

Figure 5 : Moving edges detection with classical and fuzzy approaches

5.2 Architecture Overview

The general structure of the moving edges detector is presented in figure 6. In order to perform a real time processing on 3x3 size masks, two lines delay and six data registers are used, allowing thus a simultaneous access to the nine video data of the processed mask. Two cascadable memory plans are used to store the binary images I'_n and I'_{n-1} required by Orkisz operator.

The fuzzy edges detector is constituted of a partitioning data block (figure 7), and an edges detection block (figure 8). The eight outputs of the partitioning data block {*S*i, i=1,8} are used by the same number of logic functions to examine the different configurations associated to a contour point. The eight logic functions outputs {*T*i, i=1,8} will be then used by an "OR" function to verify if at least one of the eight configurations is met. Thus, if (*Y* = 1), then the considered central point *I*'(*x*,*y*) is taken for a contour point otherwise, it is considered as belonging to a uniform zone.

Finally, Orkisz operator block is used to perform a moving edges detection with the three consecutive binary pictures (*I*'n-1, *I*'n, *I*'n+1). In this case the formulation of Orkisz operator is as follow :

$$\forall x, y R_n(x, y) = I'_n(x, y) * \overline{I'_{n-1}}(x, y) * \overline{I'_{n+1}}(x, y)$$

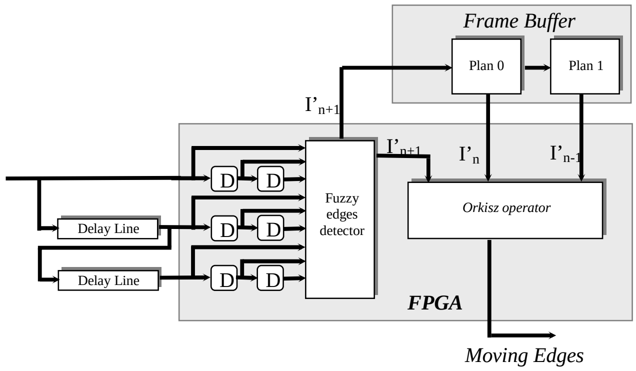

Figure 6 : General structure of the moving edges detector

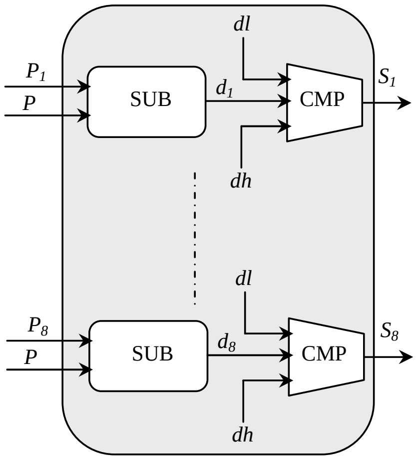

Figure 7: Data partitioning block

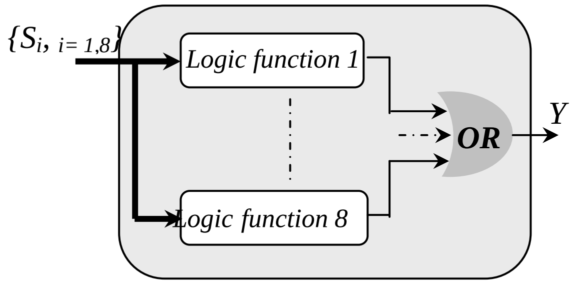

Figure 8 : Edges detection block

The use of binary pictures allow to implement easy moving edges detector and permit to have a considerable reduction of the extern memory size which remains a powerful criterion in the construction of embedded systems.

5.3 Implementation within an XC4000 FPGA

The Xilinx family of Field-Programmable Gate Arrays (FPGAs) is composed of three standard types of programmable elements : a perimeter of input/output blocks (IOBs), a core array of configurable logic blocks (CLBs), and resources for interconnection (figure 9). Xilinx FPGAs can be reprogrammed an unlimited number of times, allowing thus designers to elaborate the required hardware for a given task without having to construct new hardware for each application [10][11][12].

The devices are customised by the configuration program data stored in internal memory cells.

The XACT development system delivers a powerful software tool-set for design implementation : from schematic capture, to simulation, auto place and route, and creation of the configuration bit stream [13].

For this work we've used a real time image development system (figure 10) based on series 150/151 boards, compatible with the VME bus [14]. FPGAs which are cabled on our specific applications card use various initialisation and framing signals in order to operate in the field of validity of the video signal.

The fuzzy motion edges detector has been implemented within an FPGA XC4006 circuit and operates in real time at 10 MHz on pictures in CCIR format. The cost in surface evaluated by the number of CLBs (reconfigurable blocks) used is equal to 153 CLBs. An example of real time result obtained with our system is presented in figure 11.

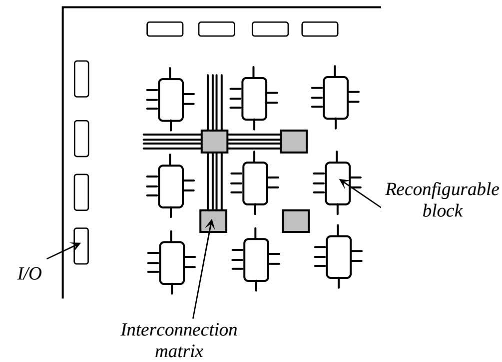

Figure 9 :. Structure of an FPGA

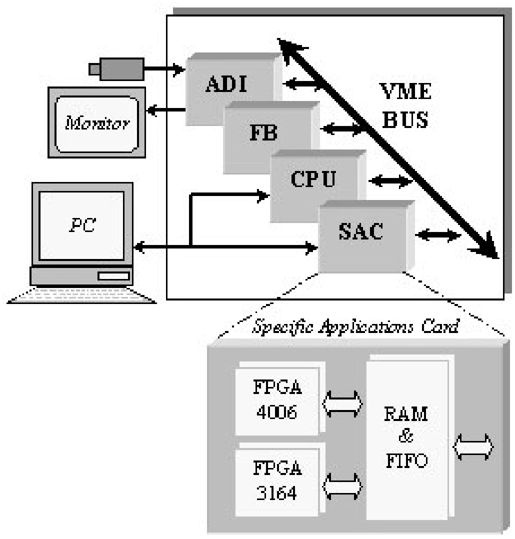

Figure 10 : Development system

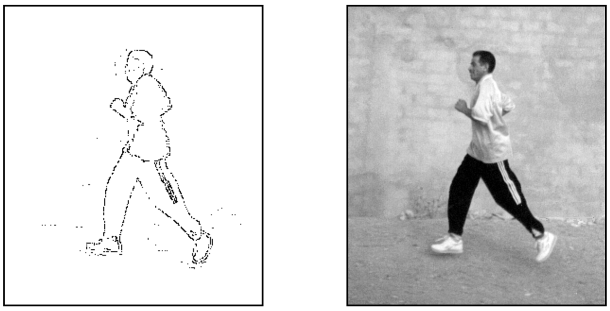

Figure 11: Moving edges in a real world scene

CONCLUSION AND PERSPECTIVES

We have presented in this article the development and the implementation of a moving objects location operator within an FPGA circuit of XILINX for a real time image processing.

It is important to note that using binary pictures allow to implement easy moving edges detectors and permit to have a considerable reduction of the extern memory size which remains a powerful criterion in the construction of embedded systems.

We have shown that the use of the fuzzy approach allows effectively to improve results obtained by the classical approach, but remains also ineffective in some particular cases.

Net improvements could be brought by using complex membership functions, more representative of the studied phenomenon but, in exchange of an increase in the realisation costs. This solution is not to exclude, because the new possibilities offered by FPGA circuits [11] and the considerable decrease of their prices allows to implement reconfigurable membership functions (using look up tables), thus permit to construct flexible architectures able to test different fuzzy operators with no further hardware expense [15].

ACKNOWLEDGMENTS

We would like to thank Professor Michel Roussel and his team from the LTI laboratory of the University of Reims (France) for their critical councils which allowed us to conclude this work.

REFERENCES

[1] Mark. S. Burrell "Computer Vision For High-Spees, High Volume Manufacturing", IEEE International Conference on Systems, Man and Cybernetics, volume 3, October 1993, pp 349 – 354

[2] S. M. Haynes, R. Jain, "Detection of moving edges" Computer Graphics Image Process, Vol 21, pp 345-367, 1983.

[3] P. Stelmaszyk, P.Bonnet, J.G. Postaire, "Analyse de scènes dynamiques par recherche des contours en mouvement", 5e congrèsReconnaissance de Formes et Intelligence Artificielle AFCET / ADI / INRIA, Grenoble (France), novembre 1985, pp 1181-1189.

[4] Ch Vieren, J.G Postaire, P. Bonnet, P. Deparis, "Détection du contour extérieur d'objets en mouvement sur fond non uniforme", Colloque GRETSI Traitement du Signal et ses applications, Juan les pains, juin 1989, pp 621-624.

[5] M. Orkisz, "Localisation d'objets mobiles dans des scènes filmées par une caméra fixe", Revue traitement du Signal, Volume 9, N° 4, 1992, pp 325-346.

[6] E. Levrat, J. Bremont, B. Heit, J. Dubois, "Edges detection using the fuzzy Set Theory", IFAC Symposium on Advanced Information Processing In Automatic Control, Nancy, 1989, Vol 2, pp 171-173.

[7] F. Russo, "A New Class of Fuzzy Operators for Image Processing", Design and Implementation, Proc Of Intl Conf on Fuzzy Systems, San Fransisco, March 28-April 1, 1993, pp 494-501.

[8] T. L. Huntsgerger, C. L. Jacobs, R. L. Cannon, "Iterative Fuzzy Image Segmentation", Pattern recognition, Vol 18, N°2, 1985, pp 131-138.

[9] G. Millon, "Contribution à l'exploitation dynamique de la reconfigurabilité des FPGAs : étude et validation d'une architecture dédiée au traitement d'image' , Thèse de Doctorat de l'Université de Reims-Champagne-Ardenne, juillet 1999.

[10] Xilinx 1994, The Programmable Gate Array Data Book, San Jose, CA 95124-3400, 1994.

[11] A. Dehon, "The Density advantage of configurable computing", IEEE on computer, April 2000, pp 41-49.

[12] J. Valls, M. Kuhlmann and K.K. Parhi, "Efficient mapping of CORDIC algorithms on FPGA", Proc. of the 2000 IEEE Workshop on Signal Processing Systems (SiPS) Design and Implementation, Lafayette, LA, October 2000, pp 336-345.

[13] XACT, Viewlogic Interface User Guide, XILINX, San Jose, CA95124-34000, 1994.

[14] Imaging Technology Incorporated Series 150/151 Boards, 1990.

[15] D. Chikouche, R.E Bekka, "Architectures rapides dynamiquement reconfigurables des filtres numériques récursifs". Revue Traitement du Signal, vol 16, n°2, 1999, pp 145 -156.