A hardware architecture for fast video object recognition using SVM and Zernike Moments

Cedric LEMAITRE[a], Johel MITERAN[a], Olivier AUBRETON[b] and Romuald MOSQUERON[a]
[a]Laboratoire Le2i (UMR CNRS 5158), Faculte Mirande, Aile H, Universite de Bourgogne, BP 47870, 21078 Dijon, France ;
[b]Laboratoire Le2i (UMR CNRS 5158), 12 rue de la Fonderie, 71200 Le Creusot, France

ABSTRACT

An architecture for fast video object recognition is proposed. This architecture is based on an approximation of feature-extraction function: Zernike moments and an approximation of a classification framework: Support Vector Machines (SVM). We review the principles of the moment-based method and the principles of the approximation method: dithering. We evaluate the performances of two moment-based methods: Hu invariants and Zernike moments. We evaluate the implementation cost of the best method. We review the principles of classification method and present the combination algorithm which consists in rejecting ambiguities in the learning set using SVM decision, before using the learning step of the hyperrectangles-based method. We present result obtained on a standard database: COIL-100. The results are evaluated regarding hardware cost as well as classification performances.

Keywords: Zernike Moments, SVM, Hyperrectangles-based method, Classifier Combination, Binary pattern, FPGA

1. INTRODUCTION

Object and pattern recognition is a classical field of study in computer vision object recognition. This kind of application is mostly composed of two main parts: a characterization step and a classification step. It exists an important number of characterization methods, in particular global[1] and interest point-based[2] methods but their major limit, for a standard software implementation on classical computer, is the computing time. Indeed Viola and Jones application[3] have a process speed of 15 frames/sec. In previous work[1], we tested Motion descriptors[4] with a speed process of 19 frames/sec. Our major aim here is to propose an architecture which permits to achieve high frequency (approximately 1000 frames/sec) and robust (invariant under image rotation, translation, scale ...) object recognition application on hardware device like FPGA[1].

Some authors have implemented classifiers using FPGA or ASIC[5, 6], allowing to obtain a very fast decision time. However, only a few fast video full object recognition methods (including feature extraction) are available in the literature. We propose here a study of a FPGA based parallel implementation of both steps. The first step, feature extraction, is based on a fast computation of Hu[7] moments and Zernike moments[8], and the second step, robust classification, is performed using a method of approximation of the SVM decision function[9-11].

This paper is composed of three main parts. In section 2 we present and compare some fast operators for object characterization. We explain why the moment-based methods are particularly interesting in the field of fast video. We compare the two best known moment-based methods used in object recognition: Hu moments and Zernike moments.

The choice of an SVM-based classifier is justified in section 3, as well as the approximation method, build using a combination of the SVM classifier learning step and a hyperrectangles-based decision function.

In section 4, we introduce the full architecture for fast video object recognition systems. In this section we estimate the necessary space for the implementation of the set: feature extraction and classification system. We also determine the maximum frequency of the full processing system.

[1] FPGA : Field Programmable Gate Array

2. INPUT FEATURES

2.1 Theory of moments in object recognition

Moments and functions of moments have been utilized as pattern features in a number of applications to achieve invariant recognition of two-dimensional image patterns. Generally a $p+q$ order moment applied on an image ($Im(x, y)$) is defined by:

$$\phi_{pq} = \sum_x \sum_y f_{pq}(x, y) \times Im(x, y) \tag{1}$$

Hu first introduced moment invariants in 1961[7], based on geometric moment. Using nonlinear combinations of geometric moments, he derived a set of invariant moments, which has the desirable properties of being invariant under image translation, scaling and rotation. However there is some information redundancy between different order invariants. There, Teague[12] suggested in 1980 the notion of orthogonal moments to recover the image from moments based on the theory of orthogonal polynomials, and introduced Zernike moments. In 1990 Khotanzad[8] first proposed using Zernike moments for pattern recognition application. In the next parts of this paper we present the method for computing a Real-Time approximation of moments and the study of Hu and Zernike moments.

Let us recall here that our final goal is to implement our process into a FPGA board. The definition of a $p+q$ order moment given by eq. (1) shows that a moment is computed by projecting a picture Im on a basis f_{pq} . This basis consists of float values which require much place for storing in an internal FPGA memory as well as an external memory. In order to reduce the hardware cost, we propose to approximate the projection basis (f_{pq}) using dithering. This function will transform float elements of the basis into binary elements. In[13, 14], Aubreton described the method to compute moment described the method to compute moment with binary projection basis. This method use the binary projection basis obtained by dithering and its negative as shown by the following equation:

$$\phi_{pq} = Max \sum \sum F_{pq}(x, y) \times \mathrm{Im}(x, y) + Min \sum \sum \overline{F_{pq}}(x, y) \times \mathrm{Im}(xy) \tag{2}$$

For computing dithering basis, we propose to use the standard Foyd-Steinberg dithering method. An example of Zernike basis and the same binary basis after dithering is shown on figure. In follows parts, we will study two standard moment-based methods: Hu invariants and Zernike Moments.

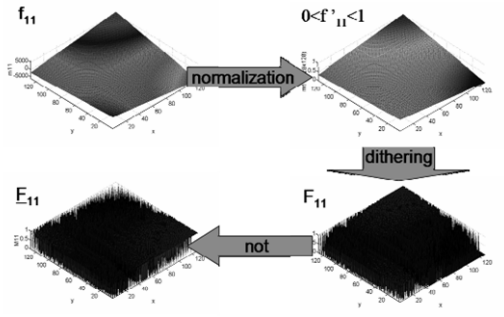


Figure 1: Different steps in the determination of the binary basis Fpq and its negative

2.2 Hu invariants

Hu invariants are based on center-geometric moments which are defined by:

$$f_{pq} = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} (x - x_0)^p (y - y_0)^q \tag{3}$$

Where x_0 and y_0 are the barycentre of the image.

Hu invariants are nonlinear combinations of center-geometric moments which have interested properties: translation and rotation invariance. The five first hu invariants are defined by:

$$\phi_1 = \mu_{20} + \mu_{02} \tag{4}$$

$$\phi_2 = (\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2 \tag{5}$$

$$\phi_3 = (\mu_{20} \times \mu_{02})^2 - \mu_{11}^2 \tag{6}$$

$$\phi_4 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{30})^2 \tag{7}$$

$$\phi_5 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{30})^2 \tag{8}$$

2.3 Zernike Moments

Zernike Moments are introduced by Teague in image analysis and form a complete orthogonal set over the interior of the unit circle $x^2 + y^2 = 1$. The Zernike function of order (p,q) is defined in the polar coordinate system (r,θ) by the following complex function:

$$f_{pq}(r,\theta) = \sum_{k=q,\, p-|k|=even}^{p} \frac{(-1)^{\frac{p-k}{2}} (\frac{p+k}{2})!}{(\frac{p-k}{2})!(\frac{q+k}{2})!(\frac{k-q}{2})!} r^k e^{(ip\theta)} \tag{9}$$

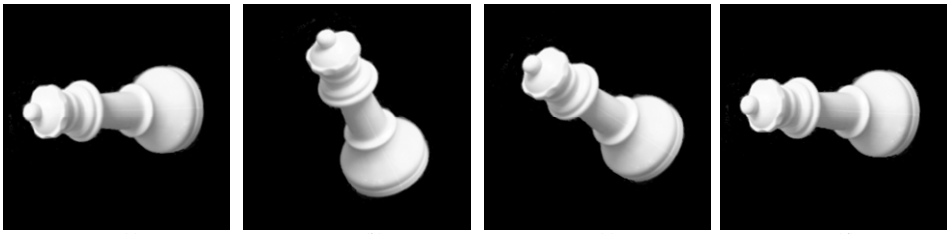
2.4 Experiments

In this section, we propose to study robustness of the two standard families of invariants which approximate by the dithering function on the test image (figure: 2). Rotation invariance and Robustness against noise are studied. For this study, we have used the approved protocol that Ghorbel used in [15].

Figure 2: Test image

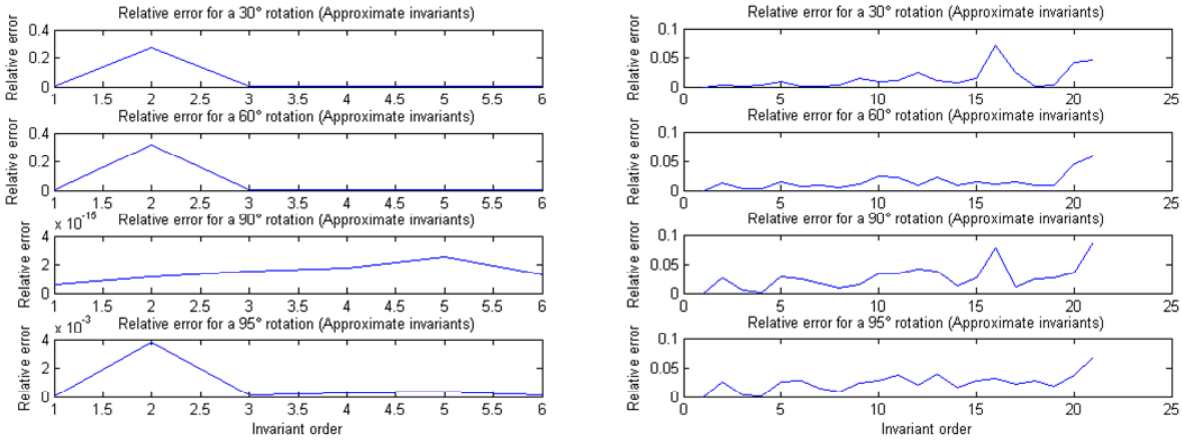
- Invariance against rotation

In order to test invariance against rotation, we have rotated the test image (fig. 2) by 30°, 60°, 90° and 95°, as shown in (fig. 3). Plots in (fig. 4) compare the relative error between the invariant vector of the original image and the invariant vectors of the rotated images. Plots clearly show that the relative error is much higher with Hu invariants than with Zernike moments. Another important point to note is that some particular Zernike moments are disturbed but less disturbed than Hu invariants.



(a) (b) (c) (d)

Figure 3: Several rotation of the original image: (30° (a), 60° (b), 90° (c), 95° (d))

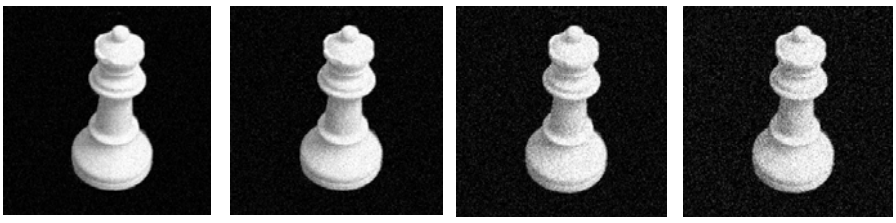


(a) (b)

Figure 4: Rotation invariance using approximated invariants: (Hu (a), Zernike (b))

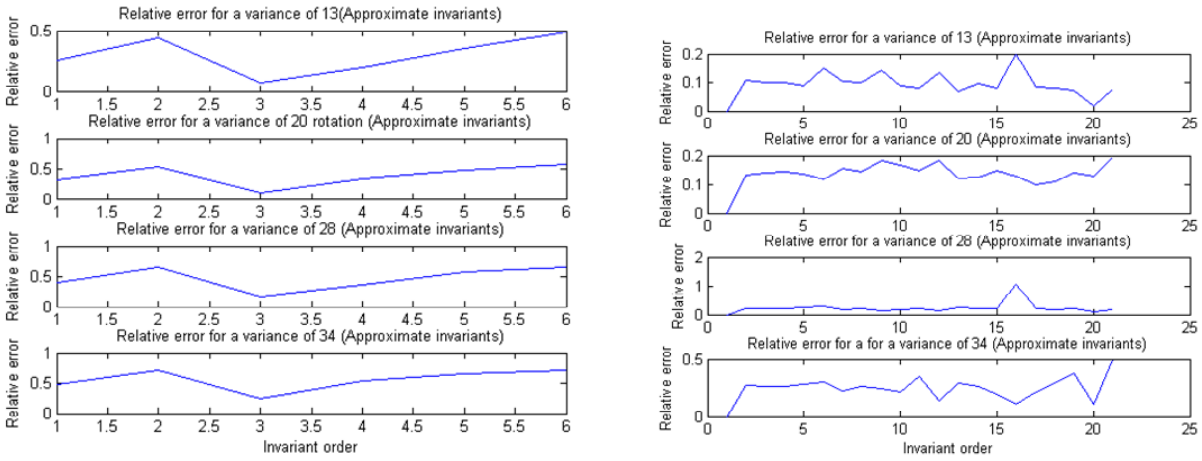
- Robustesses against noise addition

In order to test the robustness of the invariants against noise addition, we have added to the test image (fig. 2) a Gaussian noise with mean $\mu = 0$ and variance $\sigma^2 = 13,\ 20,\ 28$ and 34 as shown in (fig. 5). Plots in (fig. 6) compare the relative errors between the invariant vector of the original image and invariant vectors of the noisy images. There is no fundamental difference between the two families and, as expected, higher-order moments are the more sensitive to noise addition.



(a) (b) (c) (d)

Figure 5: Several noise variances of the original image (13 (a), 20 (b), 28 (c), 34(d))



(a) (b)

Figure 6: Robustness against noise using approximated invariants: (Hu (a), Zernike (b))

2.5 Operator Choice

Considering the fact that Zernike moment are not really disturbed by rotation and noise, considering the property that Zernike moments have no information redundancy between different orders and that the number of features is much higher in the case of Zernike moments, we have decided to chose Zernike moments for our application.

3. SUPPORT VECTOR MACHINES (SVM)

3.1 Definition

Support Vector Machines (SVM) are universal learning machines developed by Vladimir Vapnik[11] in 1979. A review of the basic principles follows, considering a 2-class problem (whatever the number of classes, it can be reduced, by a "'one-against-others"' method, to a 2-class problem).

The SVM perform a mapping of the input vectors (object) from the input space (initial features space) R_d into a high dimensional feature space Q; the mapping is determined by a kernel function K. It finds a linear (or non-linear) decision rule in the feature space Q in the form of an optimal separating boundary, which leaves the widest margin between the decision boundary and the input vector mapped into Q. This boundary is found by solving the following constrained quadratic programming problem:

Maximise

$$\mathrm{W}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i . \alpha_j . y_i . y_j . K\left(x_\mathrm{i}, x_\mathrm{j}\right) \tag{10}$$

Under the constraints

$$\sum_{i=1}^{n} \alpha_i . y_i = 0 \tag{11}$$

And $0 \le \alpha_i \le T$ for $i = 1,\ldots,n$ where $xi \in R_d$ are the training sample set vectors, and $y_i \in \{-1;+1\}$ the corresponding class label. T is a constant needed for non-separable classes. $K(u,v)$ is an inner product in the feature space Q which

may be defined as a kernel function in the input space. The condition required is that the kernel $K(u,v)$ be a symmetric function which satisfies the following general positive constraint:

$$\iint_{R_d} K(\mathbf{u},\mathbf{v})\mathrm{g}(\mathbf{u})\mathrm{g}(\mathbf{v})\mathrm{d}\,\mathbf{u}\,\mathrm{d}\,\mathbf{v} > 0 \tag{12}$$

Which is valid for all $g \neq 0$ for which

$$\int g^2(\mathbf{u})\ \mathrm{d}\mathbf{u} < \infty \tag{13}$$

The choice of the kernel $K(u,v)$ determines the structure of the feature space Q. A kernel that satisfies eq. (13) may be presented in the form:

$$K(\mathbf{u},\mathbf{v}) = \sum_k a_k \Phi_k(\mathbf{u})\Phi_k(\mathbf{v}) \tag{14}$$

Where a_k positive scalars and the functions are Φ_k represent a basis in the space Q. Vapnik three types of SVM [11]:

Polynomial SVM:

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}.\mathbf{y}+1)^p \tag{15}$$

Radial Basis Function SVM (RBF) :

$$K(\mathbf{x},\mathbf{y}) = e^{\left(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)} \tag{16}$$

Two-layer neural network SVM:

$$K(\mathbf{x},\mathbf{y}) = Tanh\{k.(\mathbf{x}.\mathbf{y}) - \Theta\} \tag{17}$$

The Kernel should be chosen a priori. Other parameters of the decision rule (17) are determined by calculating (18), i.e. the set of numerical parameters $\{\alpha_i\}_1^n$ which determines the support vectors and the scalar b.

The separating plane is constructed from those input vectors, for which $\alpha_i \neq 0$. These vectors are called support vectors and reside on the boundary margin. The number N_s of the support vectors determines the accuracy and the speed of the SVM. Mapping the separating plane back into the input space R_d, gives a separating surface which forms the following nonlinear decision rules.

$$\mathrm{C}(\mathbf{x}) = \mathrm{Sgn}\left(\sum_{i=1}^{Ns} y_i\alpha_i \cdot K(\mathbf{s}_i,\mathbf{x}) + b\right) \tag{18}$$

Where s_i belongs to the set of N_s support vectors defined in the training step.

3.2 Classication results

We propose here to compare the robustness of Zernike approximate moments in a categorization problem with the results of exact Zernike moments to categorization problem. For that we use a standard database: COIL-100[16]. We have sorted a part of this dabase into 4 classes (table 1) which contain: soda canes, rectangular boxes, mugs and cars. All pictures are 265x256 pixels. The results of classification are shown in the table 2. In this experiment, we tune the σ

value in order to optimize the classification results. Parameters of the kernel have an important role and when the error rate of approximate Zernike moments is low ($e = 1.12\%$) ; this approximation is usable in an embedded system.

Now we need to study SVM approximation to have a full application.

Table 1. Summary of the test database

Class number	Samples number	Kind of object	Image example
1	20	Rectangular boxes	
2	55	Soda canes	
3	24	Mugs	
4	67	Cars	

Table 2. Classification results

Value of σ of the RBF kernel	CV error rate for exact Zernike Moments	CV error rate for approximated Zernike Moments
0.1	7.4	10.11
1	1.8	2.21
10	0.6	1.12
100	1.8	2.8
1000	11.4	33.79

3.3 Approximation for Hardware implementation

- Method

One can see that the decision rule is easy to compute, but the cost of parallel implementation in ASIC or FPGA is clearly higher than in other methods like the hyperrectangles-based method for example. Even if the exponential function can be stored in a particular LUT in order to avoid computation, the scalar product $K(.,.)$ requires some multiplications and some additions; the final decision function requires at least one multiplication and one addition per support vector. For a given model (set of support vectors), operators can be implemented using constant values (KCM[17]) as we did in the

hyperrectangles-based method described in a previous work. However, the cost of multiplication is significantly more important than the cost of comparator. Chapman[17] proposes a structure using 20 slices[2] per 8 bit multiplier. An 8 bit adder uses 4 slices. The hardware cost of a possible SVM parallel implementation is detailed in[9] . We have estimated the number of adders and multipliers by a fully parallel computation of $K(.,.)$ and the final sum of products, in the case of a simplified RBF kernel. Given the number of slices needed by the computation of each elementary operator, we deduced λ_{svm} the hardware cost of each implementation in slices:

$$\lambda_{svm} = 72(3d-1)Ns + 8 \tag{19}$$

In order to reduce the hardware cost, a classifier combination is proposed in[9]. Combining classifiers is a classical way to increase performance of the general pattern recognition problem[18]. Three main methods are commonly used: sequential, parallel and sequential parallel. These approaches allow increased performance, but the cost of the hardware implementation is high since all the decision functions have to be computed in order to obtain the final classification.

More generally, it is possible to combine classification methods during the training step[10] as shown in figure: 7. Here we propose such a combination, allowing an approximation of SVM decision boundaries using hyperrectangles; thus the hardware cost is:

$$\lambda_h = d\sum_{z}^{y=1} m_y \tag{20}$$

Where d is the size of feature dimension space, z is the number of classes and m_y the number of hyperrectangles for class y .

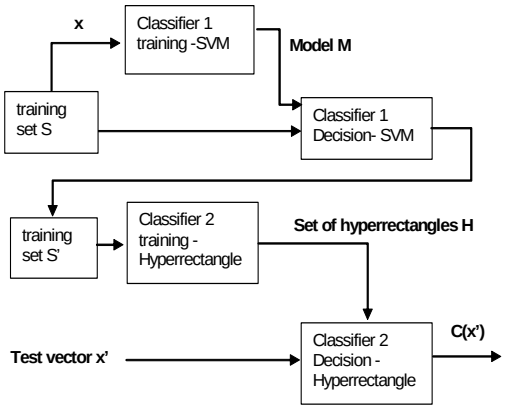


Figure 7: Combination of training steps

- Experimental results

We have tested our method using images of the database described in section : 3.2 in table : 1. As shown in table : 3, our combination method allows to obtain a error rate $e = 1.83\%$, similar to the SVM one (1.12%), using a lower number of slices($\lambda = 4864$ slices). However it is still possible to optimize the hardware cost, following the strategy defined in:

From a training set S

$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)\}$, build a model M containing support vectors using SVM algorithm :

[2]slices : base element of the FPGA structure

$$M = \{K, (s_1, y_1), (s_2, y_2), \ldots, (s_{N_s}, y_{N_s}, b\}$$

Build S', the new training set, classifying each sample of S using M according to eq.(18):

$$S' = \{(x_1, y_1'), (x_2, y_2'), \ldots, (x_p, y_p')\}$$

Build H, a set of hyperrectangles using S' and the algorithm described in[19].

Table 3. Performance of combination system

Hyperrectangles error	17.78
Number of hyperrectangles	138
SVM error (%)	1.12
Combination error (%)	1.83
Number of hyperrectangles in approximated space	76

As a conclusion, for the comparable error rate (for SVM and for the combination method), the combination method allows to obtain a decision rule with only 1208 slices against an SVM hardware cost of slices (obtain with eq. (19) where $N_s = 54$).

Table 4. Performance and hardware cost

Learning samples number	Number of slices	Error rate (%)
11	779	3.37
15	1024	2.28
19	1208	1.83

4. FINAL HARDWARE ARCHITECTURE AND EXPERIMENTAL RESULTS

In previous sections, we studied some methods which allow to approximate Zernike Moments and SVM decision rule. We have chosen approximation methods in order to minimize the hardware cost including two steps. The architecture (fig. 8) is proposed for realise high speed object recognition application. This application work as follow:

First we could recall that the function which computes Zernike moments is a projection of the input image on Zernike basis. In section 2.1, in order to reduce hardware cost in particular the number of memory access, we propose to project the input image on binary basis which obtained with the dithering function. With binary patterns, we can have a profit of memory capacity of a 100 factor (0,164 MB for binary patterns and 10.5 MB for exact pattern coded on double precision).

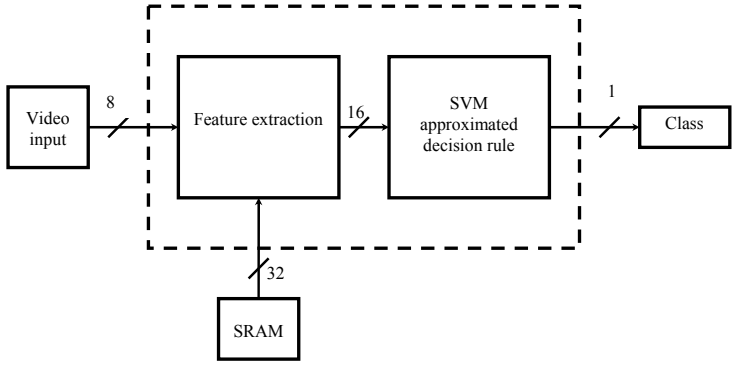


Figure 8: Global architecture

For computing 16 first approximate Zernike moments, we need store 21 binary patterns (for real and imaginary parts of Zernike polynomial) in an external 32 bit memory. To obtain approximate Zernike moments, we use the accumulation structure described by (fig. 9).

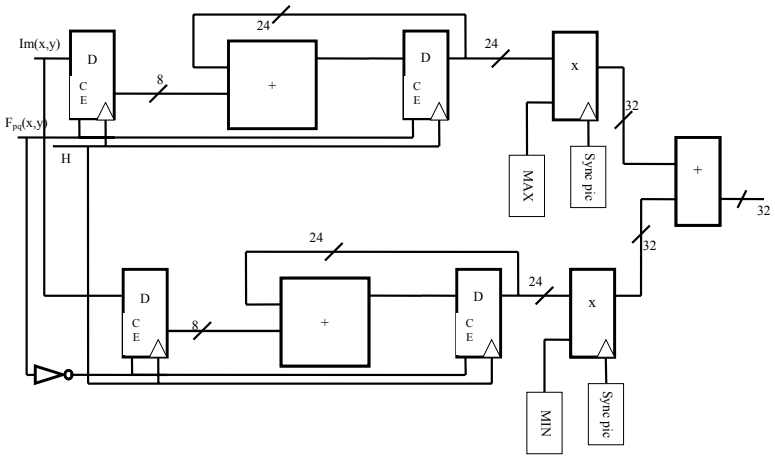


Figure 9: Feature extraction function

For all pixels, we activate the accumulation operation with an 8 bit flip-flop which was validated by binary patterns signal $F_{pq}(x,y)$. The input of flip-flop is a pixel value $IM(x,y)$ with 8 bit width. The accumulation is realized with 24 bit adder and 24 bit flip-flop. The multiplier is used in order to normalize the accumulator output.

This architecture is repeated a second time for the other part of the eq.(2). At the end of the process, we add the two parts. In the case of complex moment, we need apply this architecture for real and imaginary part before compute the modulus. In order to compute 16 approximate Zernike moments in parallel, we repeat the previous structure 21 times (Some moments have only a real part).

In order to obtain the current value of 21 binary patterns in the same times with only one memory access, we propose the memory arrangement shown in (fig. 10). After obtaining the 16 values of approximate Zernike Moments, we transmit this value to the classification function which allows obtaining the decision. The table: 5 make a summary of the different maximum delay and hardware costs. The characterization function needs a maximum delay of 23.5ns and the classification function: 18.03ns. Ideally with this architecture, we can have a process speed of 3674 frames/sec.

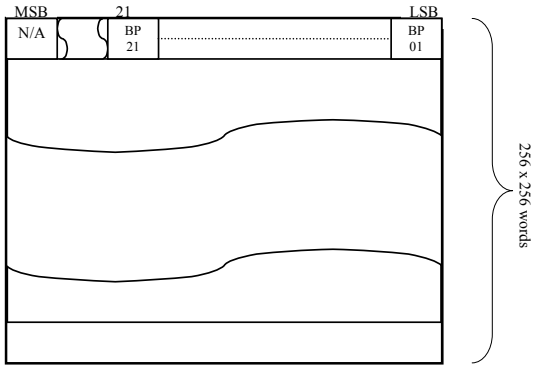


Figure 10: arrangement of binary patterns in the external RAM

Table 5. Summary of the performances on own database

	Feature extraction function	Classification function	Error rate (%)
Hardware cost (slices)	837	779	N/A
Maximum delay (*ns*)	23	18	10

5. CONCLUSION

In this paper, we have described architecture for hardware implementation to realise a fast video object recognition application. This architecture uses two improved parts of the pattern recognition field: analysis by moments for characterization and Support Vector Machines for classification. Those methods cannot be directly implemented on the hardware targets like FPGA, we need to studied some approximation methods for feature extraction and classification.

For the approximation of characterization methods approximation, we have proposed to compute invariants using moment-based function. In order to reduce the hardware cost, we have proposed to use an approximation of these moment-based functions by using a standard dithering method (Floyd-Steinberg dithering. We have evaluated two moment-based methods: Hu invariants and Zernike moments. Regarding the results, Zernike moments are more robust (invariance against rotation and noise) than Hu invariants.

In a second part, we present a method of approximation of the decision function of a powerful classifier. This method is an SVM approximation using hyperrectangles-based method which has the major advantages to have a low hardware cost. Regarding results of exact and approximate Zernike moments with our SVM approximated classifier, we could conclude that the classification error ratio is clearly acceptable for exact moments as well as approximate Zernike moments and we could use these two tools for full hardware architecture.

Finally we propose architecture for FPGA device based on approximate Zernike moments and approximate SVM. Although the classification error ration is slightly higher our two approximations allow decreasing largely the hardware cost of the system. Ideally with this architecture, it is possible to obtain approximately a process speed of 3600 frames/sec on standard existing FPGA as Xilinx Spartan or Virtex families.

As a perceptive for futher work, we may mention the implementation of the complete architecture on Xilinx Spartan-3[20] or Vitex-4[21] for example. In addition, we will test this method on larger standard databases.

ACKNOWLEGEMENT

We would like to thanks

Jean-Paul Auffrand, Departement of Languages Applied to Science and Technology, Faculté des Sciences & Techniques – Mirande, Université de Bourgogne

For his help in proof-reading this paper.

REFERENCES

[1] F. Smach, C. Lemaitre, J. Miteran, and J. P. Gauthier, "Reconnaissance d'objets 3D couleur à l'aide des descripteurs de Fourier Généralisés et des SVM," presented at Journées d'étude SEE/GRD ISIS : La reconnaissance des formes : quelles méthodes pour quelles applications?, Paris, 2006.

[2] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from Maximally Stable extremal Regions," presented at BMVC, London, 2002.

[3] P. Viola and M. Jones, "Robust Real-Time face detection," *IJCV*, vol. 57, pp. 137-154, 2004.

[4] J. P. Gauthier, G. Bornand, and M. Zilbermann, "Harmonic analysis on motion groups and their homogeneous spaces," *IEEE transaction on Systems*, vol. 21, 1991.

[5] R. R. Rojas, D. Dragomirecu, D. Houzet, and D. Esteve, "Implementation of THE SVM generalization function on FPGA," presented at ISPC, 2003.

[6] R. R. Rojas, D. Houzet, D. Dragomirecu, F. Carlier, and S. Ouadjout, "Object recognition system on chip using the support vector machines," *Journal on Applied Signal Processing*, vol. 2005, pp. 993-1004, 2005.

[7] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transaction on Information Thery*, vol. 2, pp. 179-187, 1962.

[8] A. Khotanzad and Y. Hong, "Invariant image recognition by Zernike moments," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 489-497, 1990.

[9] J. Miteran, S. Bouillant, and B. E., "SVM approximation for real-time image segmentation by using an improved hyperrectangles-based method," *Real-Time imaging*, vol. 9, pp. 179-188, 2003.

[10] B. Moobed, "Combinaison de classifieurs, une nouvelle méthode," in *Laboratoire d'informatique de polytechnique d'Orsay*. Paris: Université d'Orsay, 1996.

[11] V. Vapnik, *The nature of statistical learning theory*. New York, 1995.

[12] M. Teague, "Image analysis via the general theory of moments," *Journal of American Optic Society*, vol. 70, pp. 920-930, 1980.

[13] O. Aubreton, "Rétines à masques : Utilisation de masques binaires pour l'implantation d'un opérateur de reconnaissance de formes dans une rétine CMOS.," in *IUT Le Creusot*. Le Creusot: Burgundy university, 2004.

[14] O. Aubreton, L. F. C. Lew Yan Voon, G. Cathébras, F. Moniot, and B. Lamalle, "A CMOS retina for Zernike Moments Estimation," presented at SPIE, San Jose, USA, 2005.

[15] F. Ghorbel, S. Derrode, R. Mezhoud, T. Bannour, and S. Dhahbil, "Image reconstruction from a complete set of similarity invariants extracted from complex moments," *Pattern recognition letters*, vol. 27, pp. 1361-1369, 2006.

[16] S. Nene, S. Nayar, and H. Murase, "Columbia object image library (coil-100)," University of Columbia 1996.

[17] K. Chapman, "Constant coefficient multipliers for XC4000. Xilinx application," Xilinx 1996.

[18] J. Kittler, M. Hatlef, R. Duin, and J. Matas, "On combining classifiers," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226-239, 1996.

[19] J. Miteran, P. Gorria, and M. Robert, "Geometric classification by stress polytopes. Performances and integrations," *Traitement du signal*, vol. 11, pp. 393-407, 1994.

[20] Xilinx, "Spartan-3 capacities : http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan3_fpgas/capabilities/index.htm," 2006.

[21] Xilinx, "Virtex-4 capacities : http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex4/product_table.htm," 2006.