



**HAL**  
open science

# Optimized pre-processing input plane GPU implementation of an optical face recognition technique using a segmented phase only composite filter

Yousri Ouerhani, Maher Jridi, Ayman Alfalou, C. Brosseau

## ► To cite this version:

Yousri Ouerhani, Maher Jridi, Ayman Alfalou, C. Brosseau. Optimized pre-processing input plane GPU implementation of an optical face recognition technique using a segmented phase only composite filter. *Optics Communications*, 2012, 289, pp.33-44. hal-00782668

**HAL Id: hal-00782668**

**<https://hal.science/hal-00782668>**

Submitted on 30 Jan 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimized pre-processing input plane GPU implementation of an optical face recognition technique using a segmented phase only composite filter

Y. Ouerhani,<sup>1</sup> M. Jridi,<sup>1</sup> A. Alfalou,<sup>1,\*</sup> and C. Brosseau<sup>2</sup>

<sup>1</sup>Equipe Vision, L@BISEN, ISEN-Brest,

20 rue Cuirassé Bretagne CS 42807, 29228 Brest Cedex 2, France

<sup>2</sup>Université Européenne de Bretagne, Lab-STICC, Université de Brest,

CS 93837, 6 avenue Le Gorgeu, 29238 Brest Cedex 3, France

\*ayman.al-falou@isen.fr

**September 2<sup>nd</sup> 2012**

**Keywords:** correlators, correlation, phase-only filters, tracking, optical engineering

## **Abstract**

The key outcome of this work is to propose and validate a fast and robust correlation scheme for face recognition applications. The robustness of this fast correlator is ensured by an adapted pre-processing step for the target image allowing us to minimize the impact of its (possibly noisy and varying) amplitude spectrum information. A segmented composite filter is optimized, at the very outset of its fabrication, by weighting each reference with a specific coefficient which is proportional to the occurrence probability. A hierarchical classification procedure (called two-level decision tree learning approach) is also used in order to speed up the recognition procedure. Experimental results validating our approach are obtained with a prototype based on GPU implementation of the all-numerical correlator using the NVIDIA GPU GeForce 8400 GS processor and test samples from the Pointing Head Pose Image Database (PHPID), e.g. true recognition rates larger than 85 % with a run time lower than 120 ms have been obtained using fixed images from the PHPID, true recognition rates larger than 77 % using a real video sequence with 2 frame per second when the database contains 100 persons. Besides, it has been shown experimentally that the use of more recent GPU processor like NVIDIA-GPU Quadro FX 770M can perform the recognition of 4 frame per second with the same length of database.

# 1. Introduction

## 1.1. Context

The need for reliable technologies to facial recognition and for extraction of landmarks and features is of paramount importance for numerous government and commercial applications such as classification, access control, biometrics, and security systems [1-7]. As state-of-the-art face recognition technology becomes more complex with, for instance, security and privacy requirements, there is a need to improve on existing techniques in order to fully satisfy the latter. The current status of theoretical and experimental research is that many optical correlation techniques (OCT) [1-5] for detecting and recognizing objects are available, ranging from the mundane to the very sophisticated. The main advantage of OCT, as compared with their digital counterparts, is that they are based on a Fourier transform (FT) operation on the input image and a simple convergent lens. In spite of this proliferation of techniques, their all-optical implementations are rather complex to develop. Thus, alternate hybrid (numerical-optical using optoelectronic interface) methods have been proposed as substitutes for all-optical techniques [1-4]. These approaches, however, are not the universal panacea and have their drawbacks and stringent requirements, i.e. aberration effects, alignment of components, limitation of the overall speed by how fast the information can be updated on the input and output devices, and need of a costly optoelectronic interface. Furthermore, use of all-optics cannot be justified for many applications, especially when the target image size is small. We note also that while an all-optical approach can be well controlled, it has also severe limitations as was shown in a recent work dealing with a surveillance video system which detects various posture-based events [7], i.e. the target image size should be small and the video rate should not be excessive.

Interest in the field of correlation techniques has been recently rekindled due to their high discriminating power, their large robustness against various types of noise, and because they allow us to simultaneously identify and determine the spatial position of specific images in a scene [3-10]. Among the different kinds of correlation optimisation techniques suggested in the literature, Refs. [4-11] present a set of correlation filters adapted to their corresponding application. As was established in Ref. [10], a novel technique for face recognition based on an independent component analysis (ICA) model was suggested. While maintaining algorithmic simplicity, this approach based on ICA representation significantly increases the true recognition rate compared to that obtained with another identity recognition method based on optical correlation and a standard composite filter. A validation of the correlation method for the purpose of recognizing a 3D object has been thoroughly described in Ref. [11].

## **1.2. Hybrid implementation of correlation**

In recent years, hybrid techniques, i.e. numerical implementation of correlation, have been considered an alternative to all-optical methods because they show a good compromise between performance and simplicity. Although hybrid techniques have generated many impressive results for the problem of correlation (we discuss them further in the text), a numerical approach frequently requires some adaptation of the correlation method in order to optimize the resources of the numerical target (CPU, GPU, ...). On the other hand, an all-numerical implementation leads to a larger applicability, reconfigurability, and adaptation of the correlation pre-processing to the considered application. One of our primary motivations is to present a pre-processing step for the target image allowing us to minimize the impact of its (possibly noisy and varying) amplitude spectrum information. This will reduce the number of necessary references to deal with every rotation of faces, and so improve the robustness,

the discriminant power, and rapidity of the correlator. To achieve this, it would be highly desirable to have a correlation method which can be implemented using an all-numerical and reprogrammable target such as the graphics processor unit (GPU), or the field-programmable gate array (FPGA) [12-15].

Among numerous filter candidates, the composite filter family [3, 16] has drawn much attention. Composite filters allow merging of different reference images, thus reducing the number of filters to be used for obtaining a reliable decision and also decreasing the time for making a decision. One would like to optimize the fusion of the reference images in the Fourier plane of the composite filter. For that purpose segmented a composite filter will be used [17]. Furthermore, the segmentation criterion is adapted to deal with the occurrence probability of a given reference at the system input, e.g. the reference face with null angle of rotation is the most probable since a subject willing to enter into a room should look in front of the camera. Decrease of the memory size to store the different filters and the time needed to manipulate them can be also achieved. An interesting example is the case of phase-only filters (POF) [18] since one gets rid of the filter for spectral magnitude which requires large bit-codes.

### **1.3. Outline**

Our study focuses on an all-numerical implementation of a robust and discriminating correlator for facial identification. Its architecture was optimized for CUDA-enabled GPUs. This implementation enables dramatic decreases in computing run time of correlation. Thus, iterative pre-processing can be used with segmented filter to increase correlator performances for making a decision.

In the remainder of this paper, we present a new procedure that allows us to detect a face in a given scene. The main novelty of the approach relies in (i) the specific pre-processing optimization step, based only on the phase information in the spectral domain [20] in the input plane, (ii) the use of a composite filter optimized for face recognition, and (iii) the use of a hierarchical classification procedure adapted for GPU processor based on two-level decision tree (DT) learning approach. On the first level, a classifier sorts out among the different classes of the learning base. On the second level, we deal only with classes having the maximum likelihood with the target image according a specific criterion. Then, a final comparison of the various winning class is made to determine the target image. We pay special attention to the choice of the decision criterion to be used in the correlation plane. An important parameter, the receiver operating characteristic (ROC) curve, which reflects the selection of possibly optimal models and discards suboptimal ones independently from (and prior to specifying) the cost context or the class distribution, has been calculated for each case considered [20] This allows avoidance of the limitations of the peak-to-correlation-energy criterion which is extremely noise-sensitive.

The rest of the paper is organized as follows: In section 2 we present and discuss the algorithmic formulation of correlation method. A key component of this section is to present the basic principle of the all-numerical implementation, justify the use of POFs in the algorithm, then the optimized pre-processing and the adaptation of the segmentation criterion, and finally discuss the decision criterion employed in this study. In section 3 we introduce the validation of an optimized algorithm for face recognition. Furthermore, we discuss the problem of optimizing correlation filters and the implementation of two-level DT-based procedure. We also illustrate this procedure with some experimental examples. In section 4 we finish with a short summary.

## 2. Optimized correlator

Here we offer a glimpse of the practical design of the correlator in order to render it more robust, faster, and adapted to an all-numerical implementation. A first optimization step involves introducing a pre-processing which is specific to the target images. The optimization method is an iterative one using the spectral phase of the target image in order to reconstruct the target image. We have evaluated the robustness of the correlator by varying our input parameters such as the rotation of the target face and different types of noise input added in the input plane. We also want to study the ability of the segmented composite filter to merge the different reference images and test a specific segmentation criterion. As a starting point for our consideration, we discuss the correlation principle from the algorithmic point of view. Our approach consists in describing the building blocks of the correlation method within the context of an all-numerical fast approach.

### 2.1. Correlation algorithm

The schematic diagram of the basic used algorithm is shown in Fig. 1. The basic configuration is the same as a Vander Lugt correlator [3]. The correlation algorithm is performed in seven steps in a precise order. Firstly, the scene target in which the face target is embedded is loaded on the input plane. The scene target originates either from a numerical file, or can be obtained from a digital camera. Secondly, the scene target is fast Fourier transformed (FFT). Thirdly, the target image spectrum is multiplied by the correlation filter (off-line POF fabricated, in the fourth step, from a reference image). Fifthly, the product of the correlation filter and the input scene are inverse fast Fourier transformed ( $\text{FFT}^{-1}$ ). Sixthly, the peak-to-correlation energy (PCE, defined as the energy of the peak correlation normalized to the total energy of the

correlation plane) is calculated in the correlation plane. In the seventh step, the PCE value is compared to a given threshold ( $T$  in Fig. 1). If the PCE value is larger than the threshold, the algorithm is stopped and decision is taken that the target image is identical to the reference image used for POF fabrication. Otherwise, another correlation filter is loaded. Steps 4 to 7 are iterated until the target object is recognized or when every filter has been considered. It should be noted that the threshold is taken such as to minimize false alarms.

## **2.2. An example of database**

The example chosen for illustration of our method is to recognize and identify a target face at the correlator entrance. In order to be useful, the identification should be discriminating and robust. Such robustness is required to take into account the changes (due mainly to rotation, scale, and background noise) of the target face with respect to the reference images of the learning base. To give an example for validating our procedure only rotation is dealt with in the numerical implementation of the correlator. As mentioned previously, we considered images from the Pointing Head Pose Image Database (PHPID) [22] and selected 8 different faces shown in Fig. 2 (a). Each subject's head experiences vertical (within the  $[-15^\circ, 15^\circ]$  range) and horizontal rotations (within the  $[-45^\circ, 45^\circ]$  range) as displayed in Fig. 2(b).

## **2.3. POF**

In their seminal paper, Oppenheim and Lin [23] came up with the idea that one can get insight into optical processing with coherent light by studying just the phase component of the spectrum of an image. The theoretical approaches in correlation of images have, naturally,

focused on this property. Hence, Horner and Gianino [18] suggested a correlation filter based on the phase of a single reference image

$$H_{POF}(\mu, \nu) = \frac{R^*(\mu, \nu)}{|R(\mu, \nu)|} = \exp[-i\theta(\mu, \nu)] \quad , \quad (1)$$

where  $H_{POF}$  defines the POF manufactured with one reference image  $r$ ,  $R$  denotes the spectrum of the reference image  $r$ ,  $R^*$  is the conjugate of  $R$ ,  $\mu$  and  $\nu$  are the spatial frequencies coordinates,  $|R|$  is the module of the spectrum  $R$ , and  $\theta$  denotes the spectrum phase of  $R$ .

The main feature of the POF is to increase the optical efficiency. Besides the ability to get very narrow correlation peaks, POF has the capacity for discriminating objects. This is a convenient filter for an all-numerical approach since its magnitude, which is responsible for broadening and deterioration of the correlation peaks, does not appear in Eq. (1) [24]. Consequently, coding only the phase component requires very low memory space. Interestingly, the number of filters that can be manipulated is increased. Note that a binarized version of the POF [3], i.e. the binary POF (BPOF), has been proposed [3]. However, as is well known the POF (and BPOF) are very sensitive to even small changes in rotation, scale and noise contained in the target images [3]. For improved correlation discrimination, however, the proposed correlator makes its decision by direct action on its input plane. From [18], it can be inferred that a convenient pre-processing is based on the use of the phase target spectrum image information to de-noise the input scene and enhance the edge in the input plane. This leads to a narrower correlation peak at the output plane. This pre-processing procedure is developed below.

## 2.4. Pre-processing in the input plane

In this study we introduce a numerical pre-processing step in order to increase the discriminating power of the correlator while keeping its robustness against target image change due to face rotation and noise. The basic reconstruction procedure of the target image is a compromise between the two cases: in the first one, the sole phase in the Fourier plane is used, i.e.  $\exp i \Phi_{target}$ , and in the second one the spectral amplitude and the phase of the target image are used, i.e.  $\rho_{target} \exp i \Phi_{target}$ , where  $\rho_{target}$  and  $\Phi_{target}$  denote respectively the spectrum amplitude and the spectrum phase of the target image (the image to be recognized). For our purpose, we consider a POF filter ( $H_{pof}$ ) in the Fourier plane of the correlator, i.e.  $H_{pof} = \rho_{reference} \exp i \Phi_{reference}$  where  $\rho_{reference}$  and  $\Phi_{reference}$  denote the amplitude and phase spectrum of the considered image reference. It is important to point out that, in the first case, we obtain a Dirac delta function ( $\delta$ ) in the correlation plane if the target image is identical to the reference image

$$Corr = FT^{-1} \left[ \exp i \Phi_{target} \exp -i \Phi_{reference} \right] = FT^{-1} 1 = \delta, \quad (2)$$

i.e. leading to a sharp and large amplitude correlation peak. However, in this case the correlation is very sensitive to noise and rotation effects. We suggest adding to this phase spectrum an amplitude, i.e.  $\rho_{reconstructed} \exp i \Phi_{target}$  ( $\rho_{reconstructed}$  is the amplitude of the reconstructed image using the algorithm shown in Fig. 3), reconstructed in such a way as to achieve improved correlation discrimination (Fig. 3).

To test the robustness of our algorithm with respect to the correlator's performance, consider the schematic picture shown in Fig. 3. Our approach is to (1) select the target image contours by using only its phase component, (2) minimize the impact of the original

amplitude of the target image, i.e. content smoothing, with respect to decision making, and (3) reconstruct the input plane with an iterative method using the POF spectrum information with minimal iteration [20, 24].

To illustrate the principle of our algorithm consider the pre-processing of a single real target image of size  $N \times N$  pixels. The first step consists to place the target image  $I_c$  ( $N \times N$  pixels) into an image  $I$  ( $2N \times 2N$  pixels), i.e.  $I_c$  is surrounded by zeros (Fig. 3). Then, the FT of  $I(2N \times 2N)$  is realized. At this stage the spectrum is given by  $\rho_{target} \exp i \Phi_{target}$ . Then, the spectral amplitude is vanished:  $\rho_{target} = 1$ . Next, the  $FT^{-1}$  (inverse FT) is done and the  $2N \times 2N$  reconstructed image is obtained:  $I_{reconstructed}$  ( $2N \times 2N$ ). This contour image is encoded via  $N \times N$  pixels; other pixels representing noise. Selection of the reconstructed image ( $N \times N$  pixels) allows us to obtain the modified target image. Otherwise, a FT of the reconstructed image is realized after having placed it in a ( $2N \times 2N$  pixels) image. Then, we carried out a FT in order to obtain the reconstructed image spectrum  $\rho_{reconstructed} \exp i \Phi_{reconstructed}$  ( $\rho_{reconstructed}$  is the spectrum amplitude of the reconstructed image and  $\Phi_{reconstructed}$  is its phase spectrum). Next, the spectral amplitude is obtained ( $\rho_{reconstructed}$ ) and combined to the original phase target image spectrum ( $\Phi_{target}$ ) to obtain  $\rho_{reconstructed} \exp i \Phi_{target}$ . The algorithm is stopped when the iteration number  $N_{it}$  is equal to a given number  $n$ .

We now show simulation results. Fig. 4 (a) shows the original target image. The reconstructed target image using our algorithm with only 4 iterations is shown in Fig. 4 (b). Fig. 4 (d) shows the reference image used to fabricate the POF. For illustration, the correlation planes obtained respectively without iterations, with 4 iterations, and with 200 iterations are displayed in Fig. 4 (e), Fig. 4 (f), and Fig. 4 (g). It is important to point out the improved correlation discrimination for only 4 iterations, i.e. the POF based correlator is

capable of producing correlation peaks which are narrower than those of the original target image and also noiseless. However our determination of the correlation planes (Fig. 4 (e) and Fig. 4 (g)) shows up unexpected similarities. It is seen in Fig. 4 (c) that the reconstructed image following 200 iterations is very similar to the target image shown in Fig. 4 (a). These qualitative remarks can be placed on a more quantitative footing by calculating the mean-square error (MSE), i.e. the cumulative squared error between the reconstructed and the original target image, versus the iteration number. This illustrated in Fig. 4 (h). As can be seen the larger the number of iterations the closer the reconstructed image is to the original one. We first look at the peak-to-correlation energy (PCE), defined as the energy in the correlation peak/overall energy in the correlation plane ratio [25], in Fig. 4 (i). As can be seen, there is a maximum for  $n$  close to 4 indicating that it is useless to realize more iterations since the correlation discrimination will be smaller and the corresponding computing time will be higher i.e. this would result in slowing down the decision making of the correlator.

To test the significance of the pre-processing step on the correlator's robustness the rotation angle of the subject's face was varied. We compared the PCE values with and without pre-processing of the target image. Table 1 shows the results for rotation angles  $\theta$  ranging from  $-30^\circ$  to  $30^\circ$ . Table 1 shows clearly that the pre-processing improves the PCE values. The proposed correlator is less sensitive to rotation effects. Another way of looking at our results is to consider receiver operating characteristic curves (ROC) [21]. Fig. (5) shows the ROC obtained with the base of facial images displayed in Fig. 2 (b) and a 3-reference segmented filter [17]. The results indicate that using the POF in conjunction with pre-processing gives superior correlation discrimination. In this case, the true positive rate (TPR) is 58.8% with a false positive rate (FPR) close to 5% compared to a true positive rate of 47.2% without the pre-processing step.

Fig. 6 illustrates an example of correlation after the pre-processing step of the algorithm. The target image ( $256 \times 256$  pixels) is a noisy face Fig. 6 (a). The POF was fabricated using the reference image shown in Fig. 6 (b). POF was selected to demonstrate the increase of the decisional performance of the correlator with this kind of algorithm. In what follows, a more complex and optimized filter belonging to the POF family will be used. The autocorrelation plane obtained between the target image without pre-processing and the reference image is shown in Fig. 6 (c). The PCE is equal to 0.09. Fig. 6 (d) shows the correlation plane when the pre-processing step is used and four iterations [24]. We find that the PCE=0.86. This clearly demonstrates that the correlation discrimination quality is much better than that obtained without pre-processing. In addition, noise is reduced significantly. A similar analysis but with different types of binary and grey scale images, with and without noise, was done. In all cases, good results were obtained.

### **3. All-numerical implementation of the correlation method for face recognition**

#### **3.1. Optimized correlation algorithm**

A first optimization step consists in decreasing the number of correlation filters in the learning base, the number of reference images being kept constant. This can be done using a segmented POF (SPOF) in a two-step method [17]. First, a segmentation of the spectral plane of the correlation filter into several independent regions is realized. Second, each region is assigned to a single reference. This assignment is done according to a specific energy criterion which compares the energy (normalized by the total energy of the spectrum) for each frequency of a given reference to the corresponding energies of another reference. In practice,

we observe that this criterion can have limited utility when the reference images are relatively similar, e.g. two images of the same subject differing from a rotation angle of  $10^\circ$ . In that case, we assume that the plane of the filter is divided into many small regions. Then, each region is affected to a single reference but this operation does not lead to correlation at the output. To overcome this limitation, we proceeded to impose coefficients  $a_i$  on the chosen criterion. In all calculations, it is assumed that the pixel is assigned to one of the references if its relative energy is  $a_i$  times greater than the other relative energies. Otherwise this pixel is assigned to the major reference around this pixel. We would like to note that these coefficients are obtained iteratively by increasing the true detection rate and decreasing the false detection rate as is usually done for the composite filter. Assignment of a region to one of the two references is done according this criterion. Hence, the SPOF contains frequencies with the largest energy. As a technical point, we remark that the time to make a decision was reduced by a factor of  $M$ ,  $M$  being the number of reference images contained in the SPOF.

The database (Fig. 2) was correlated with an optimized segmented 3-reference filter. Fig. 7(a) demonstrates the good performances of this optimization scheme since the TPR is equal to 70% for a FPR less than 5 %. In addition, Fig. 7 (b) presents our results with the two optimizations contained in our algorithm. This data contained in this figure are important since they were compared with those obtained with a 5-reference optimum trade-off maximum average correlation height (OT-MACH) filter [26] taking  $\alpha=0.25$ ,  $\beta=1$ ,  $\gamma=0.1$ , and  $m_x=1$ . The optimal trade-off MACH filter can be written as

$$H_{OTMACH} = \frac{m_x^*}{\alpha C + \beta D_x + \gamma S_x},$$

where  $m_x$  is the average of the training image vectors,  $C$  is the diagonal power spectral density matrix of the additive input noise,  $D_x$  is the diagonal average power spectral density of the training images,  $S_x$  denotes the similarity matrix of the training images, and  $\alpha$ ,  $\beta$ ,  $\gamma$  are three numerical coefficients. Fig. 10 shows the ROC curve obtained after correlation of the faces

displayed in Fig. 2 with the OT-MACH filter fabricated with subject 1 and the five references corresponding respectively to angles set to  $-45^\circ, -30^\circ, -15^\circ, 15^\circ, 45^\circ$ . The comparison of Fig. 7 and Fig. 8 demonstrates that the correlation performances obtained by the optimized segmented 3-reference filter are greater than those obtained with the 5-reference OT-MACH filter.

To further decrease this time, we suggest a two-level decision tree learning approach of the correlation algorithm. The first level allows us to determine to which class belongs the target image. Next, the target image is identified among the references of the class considered. In other words, this optimization step has the purpose of avoiding comparison of the target image with every image of the learning base.

We consider a learning base which has  $N_{im}$  images (Fig. 9 (a)). In our calculations we have retained 8 subjects and 21 references were necessary to recognize the subject. Hence,  $N_{im} = 8 \times 21 + 168$ . We first group the reference images of a given subject in a class (Fig. 9 (c)). Next, we group the images of a given class according specific characteristics such as rotation (Fig. 9 (b)).

Let us examine the first level of our algorithm (Fig. 10 (a)). Basically, it is used to determine to which class of the learning base the target image belongs. The pre-processed input image is placed in the numerical correlator. After a first FT, its spectrum is multiplied by a segmented filter called filter-class  $F_c(i)$  (with reference to Fig. 10 (a)). This correlation SPOF is fabricated by the merging of different reference images of class (i). Since we are interested only in rotation of the faces five references were chosen for this study. It turns out that the number of reference images should not be above 10, otherwise a second filter per class is needed. It is worth noting that the multiplication with the correlation filter, the inverse FT, and the PCE calculation are iterated until correlation of the target image has been realized

with all classes  $\mathcal{C} \in \mathbb{I}_8$ . The results are then organized into a hierarchy according the PCE values and only classes with the largest PCE are considered.

Our correlation results using the first level of the algorithm are shown in Fig. 11, where we have plotted the different values of PCE for a given class versus the target image position in the learning base. The most noteworthy feature is that subject 5 has generally the largest values of PCE, except for two cases where it is ranked third and fourth, i.e. positions 14 and 15 shown in Fig. 11. Therefore, we will restrict our attention to four classes at most. To summarize, the purpose of the first level of the algorithm is to determine the possible class of the target image among the different classes of the learning base, i.e. we select only 3 classes looking like the target image.

Turn now to the second level of our algorithm (see Fig. 10 (b)). Different sector filters are fabricated for each class using the SPOF filter. For that purpose, SPOF was employed with 3 reference images for defining each sector (Fig. 9 (b) and Fig. 10 (b)). This requires 7 filters per class. Next, the correlation of the target image with the different sector filters is realized. Correlation results with the second level of the algorithm are shown in Fig. 12. Only 3 classes at the output of the first level step were considered.

Two plots are shown in Fig. 12 corresponding to subject 5's images placed as input. The solid curve corresponds to the correct identification, while the dashed curve shows the maximal PCE values which are obtained with the 2 other classes (false recognition). We can see a remarkable result: with only 3 classes at the output of the first level of the algorithm we obtain a recognition rate  $\approx 95\%$  and a false alarm rate close to 0%, i.e. the face is always identified as that of subject 5, or not recognized.

Several comments are in order. Three cases corresponding to 2, 3, and 4 classes were considered for understanding the first level classifier effect. The entire learning base containing all positions was concerned. The more classes there are at the output of the first

level the more correlations should be done, and thus the larger is the run time. An immediate question is: does the increase of the number of classes at the output of the first level imply an increase of the decisional performances of the correlator? The answer to this question is shown in Table 2 which summarizes our results. The first line represents the number of references (3 or 5) used in each sector filter. The second line considers the number of classes at the output of the first level. The third line is the recognition rate averaged over the entire learning base. The fourth line is the maximum run time needed to identify the face among the entire training database. As illustrated in the table 2, the recognition rate is  $\approx 74\%$  for the case of the 3-reference segmented composite filter. The recognition rate increases with the number of classes at the output of the first level. However, the corresponding run time also increases. Similar conclusions are made for the case of 5 references using in the SPOF. In addition to these observations, we note that passing from 2 classes to 3 classes at the output of the first level has for effect to increase significantly the recognition rate. However, while such an increase is not much visible beyond 3 classes, the run time is much larger. For all these reasons, we suggest that 3 classes represent a good compromise between the recognition rate and the run time of our application.

### **3.2. GPU programming using CUDA**

Today, CPU frequency no longer follows Moore's law due to the increase of the CPU consumed energy and physical limitation [27]. To cope with this problem, many researchers have proposed to use parallel processors. One interesting solution using multiprocessors for graphical processing is GPU, originally designed for computing 3D functions, i.e. lighting effects, object transformations, and 3D motion [28]. Today, GPU is used for intensive

computing and can be considered as multiple cores with a software layer that allows parallel computing. Contrary to CPU, the state of the art of GPU shows that the performances in terms of execution time are in constant evolution. Indeed, we extended the study of [29] to estimate the performances of GPU and CPU. The results of this study are summarized in Fig. 13 which shows that in 2010 the maximum operating frequency of the GPU is 15 times higher than that of CPU. This means that image processing through GPU is faster than on CPU. The computing speed is measured by GFLOPS which is equal to  $10^9$  floating-point arithmetic operations per second. The GPU family used in this work is NVIDIA due to its large adoption in modern computing systems.

From an architectural point of view, GPU is composed of one or several multiprocessors called stream multiprocessor (SM), containing several physical devices, and a global memory. As it is shown in Fig. 14, each SM is composed of many graphic processors called stream processor (SP), a memory, an instruction unit, registers, and cache memories. The memory of each SM is shared between all graphic processors. The instruction unit is a scheduler that dispatches instruction and data to be executed and saved on cache memories (constant and texture). Each graphic processor is associated to its own registers. For example, the first GPU card used in this work is the Nvidia 8400 GS graphics card which has a single SM with 8 graphic processors as illustrated in Fig. 14 (a). The second one has 4 SM where each SM is composed of 8 SP as mentioned in Fig. 14 (b). More generally, each multiprocessor is based on single input multiple data (SIMD) architecture, to perform intensive computation of highly parallelizable algorithms. CUDA [30] allows us to program the GPU. In the CUDA framework, a given sequence of instructions that should be applied to different data sets in parallel is called a kernel. Each kernel is processed by parallel threads. For reasons of addressing, parallelization, and run time control different threads are grouped together as blocks (Fig. 15).

The execution of a program on GPU is divided into two parts: host and device parts. The host part represents the software running on the host PC and used to control the GPU. The host starts program execution and copies data from CPU working memory to the global memory of the graphics card. The kernels represent the code program that runs on GPU. Each grid of the kernel is processed by the multiprocessors such that many blocks are executed in parallel. Thus, each block runs on a separate multiprocessor. Blocks are split into warps representing multiple threads. The threads of one block use the shared memory together whereas each thread is assigned its own registers and its own local memory. During one clock cycle, each graphic processor executes one thread. Consequently, the run time in terms of clock cycle number of algorithms on GPU is determined by

$$ExecutionTime = \frac{Number\ of\ threads}{Number\ of\ processors} \quad (3)$$

### 3.3. Performance using the PHPID database

Here, the focus is to demonstrate the significant performance improvement of the GPU compared to what can be obtained with standard CPUs, i.e. Pentium Dual Core CPU 2.50 GHz, Intel-Core 2 CPU 2.00 GHz. We considered 1000 images from PHPID [22]. Experimental results were performed using a prototype based on GPU implementation of the all-numerical correlator using a GeForce 8400 GS processor and a Quadro FX 770M. We also provide a further motivation by comparing our results with those from a hybrid method combining a GPU (GeForce 8400 GS processor) to implement steps 2, 4 and 5 of Fig. 1 and a CPU (Pentium Dual Core CPU 2.50 GHz) to realize steps 6 and 7 of Fig. 1. A like-for-like approach is applied to the comparisons of the run times of the different architectures. The general trend shown in Fig. 16 (a) is a linear increase of the run time with the increase of the

correlation filter position in the database. Fig. 16 (b), which represents the results obtained for only 15 filters, shows the speedup of the GPU for a small database.

As shown in Fig. 16 our face recognition algorithm run on GPU achieves impressive speedup over CPU method. We do not see an important change with the hybrid CPU-GPU method, which means that making a decision with the CPU part can be useful. The time which is necessary to make a decision with GPU architecture is less rapidly increasing than for the CPU architectures. The impact of the image size on the run time is considered in Fig. 17. These results demonstrate clearly the robustness of the GPU architecture.

### **3.4. GPU implementation of the optimized correlation algorithm**

Here, we present details of the CPU-GPU implementation of the optimized correlation algorithm shown in Fig. 10. The input plane is composed by fixed images or video data which are stored in the working memory of the host CPU. For each image, we apply the pre-processing algorithm described in Sec. 2.4. This algorithm is computed using CPU. Next, we define a region of interest on each image. This region should contain the face of the person to identify. The main objective of this selection is to reduce the size of image. This reduced image is then transferred to the global memory of the GPU in order to apply the correlation algorithm (Fig. 10). The first step of this algorithm consists to perform a FT of the input image. This is done in GPU using the CUFFT library [31]. This algorithm is divided into many threads grouped into warps as mentioned earlier. The obtained spectrum is stored in the global memory of the GPU. The second step consists in multiplying the obtained spectrum and the offline manufactured composite filter, and the result is stored in the global memory of the GPU. Each multiplication will be associated with a thread and all threads are executed

separately. Finally, a  $FT^{-1}$  is performed to the result of this multiplication, and the correlation plane is stored in the global memory.

### **3.5. Experimental real-time validation**

To the best of our knowledge our study is the first work devoted to video processing on GPU of a real-time face recognition system implementing numerically a correlator using an optimized composite filter. Computer video-based face recognition can be processed in two steps. The first one can be realized offline. The objective is to construct the database of the segmented filters for the recognition application. Three angles are considered for each position. To optimize the ratio efficiency of detection versus execution time, one segmented filter is designed using the three angles in the same time. Hence, as shown in Fig. 18 for one subject, three filters, i.e. three positions per person, are considered.

In the second phase of our face recognition algorithm, the OpenCV library is used as the interface between the CPU and the webcam. Then the “.avi” video file is split into frames with  $288 \times 352$  size. Each frame is transferred to the GPU for correlation.

The validation of our architecture was done by considering a situation for which we are trying to recognize a subject belonging to a 3-subject base and moving in front of the camera. The subject can turn his face in a plane parallel to that of the camera from the left to the right ( $-30^\circ$  to  $30^\circ$ ) with no change of expression. The video sequence contains 140 frames and lasts 6s (see Supplementary Material [32] for details). We were able to demonstrate that both of the GPU used in this work (GeForce 8400 GS and Quadro FX 770M) is able to compute in real-time each frame of the video with a rate of recognition equal to 77%. Using our algorithm, each frame of the video sequence needs a recognition time of 58.8 ms. Each

picture requires 3+9 correlations, i.e. 3 correlations for the first level and 9 correlations for the second level (3 classes with 3 filters per class).

Each correlation requires 4.9 ms, hence we need  $4.9 \times (3 + 9) = 58.8 \text{ ms}$  to recognize the face of one person in the video sequence. Therefore, for an efficient real-time recognition the frame rate should be lower than  $\frac{1 \text{ s}}{58.8 \text{ ms}} = 17.006$ . For a larger size data base composed of  $p$  persons,

the maximum number of frame per second ( $Fps$ ) computed by our GPU is given by Eq. (4)

$$Fps = \frac{10^3}{T_{corr} \times (p + 9)} \quad (4)$$

Where  $T_{corr}$  represents the computing time of one correlation and  $p$  is the number of person existing in the database. It is noted that the value of the  $Fps$  will decrease from 10.6 to 2 if  $p$  is increased from 10 to 100. Accordingly, when using the NVIDIA-GPU Quadro FX 770M, we find that  $T_{corr}$  is equal to 2.8 ms. Consequently, the maximum frame rate is equal to 29, 19 and 4 for respectively 3, 10 and 100 persons on the database.

#### 4. Summary

To summarize, we have developed an all-numerical correlator for face recognition applications. This paper presents the techniques and design methodology for the full utilization of a GPU's computational resources to accelerate the computation-intensive part of our face recognition algorithm. We have presented and validated an optimization protocol of a digital correlator by suggesting a two-level decision tree learning approach: a first level is employed to class the target image and a second level is used for identification. Recognition

rate larger than 85 % with run time lower than 120 ms have been observed for test samples from the PHPID and implementation of the digital correlator using a GeForce 8400 GS processor from NVIDIA. Regarding the video processing, we have experimentally demonstrated that we are able to perform the recognition with a frame rate of 2 and 4 for respectively GeForce 8400 GS and Quadro FX 770M when the database contains 100 persons. Finally, we believe the proposed scheme promises to provide a powerful tool for investigating face recognition problems thanks to the calculation performances and programming ease of the GPU.

### **Acknowledgment**

This research was supported by a grant from the Interface Concept Company. Lab-STICC is Unité Mixte de Recherche CNRS 3192.

## References

1. F.T.S. Yu, S. Jutamulia, Cambridge University Press (1998).
2. J.L. Tribillon, Teknéa, Toulouse, (1999).
3. A. Alfalou and C. Brosseau, in: Face Recognition, M Oravec (Eds.), Understanding Correlation Techniques for Face Recognition: From Basics to Applications, In-Tech., 2010, pp. 354-380.
4. X. Zeng, T. Inoue, N. Fukuchi, and J. Bai, Opt. Express **19** (2011), 12594-12604.
5. P. Hennings, J. Thorton, J. Kovacevic, and B.V.K. Vijaya Kumar, Appl. Opt. **44** (2005), 637-646.
6. S. C. Kim, S. C. Park, and E.S. Kim, Appl. Opt. **48** (2009), H95-H104.
7. M. Elbouz, A. Alfalou, and C. Brosseau, Opt. Eng. **50** (2011), 067003.
8. V. H. Diaz-Ramirez, Opt. Lasers Eng. **48** (2010), 1153.
9. A. A. S. Awwal, Appl. Opt. **49** (2010), B40-B50.
10. A. Alfalou and C. Brosseau, Opt. Lett. **36** (2011), 645-647.
11. M. Cho, A. Mahalanobis, and B. Javidi, Opt. Lett. **36** (2011), 861-863.
12. B. Cope, P. Y. K. Cheung, W. Luk, and L. Howes, IEEE Trans. on Computers **59** (2010), 433-448.
13. G. Poli and J. H. Saito, in: Face Recognition, M Oravec (Eds.), Parallel face recognition processing using neocognitron neural network and GPU with CUDA high performance architecture”, In-Tech, 2010, pp. 381-404.
14. I. Kuon, R. Tessier, and J. Rose, Foundations and Trends in Electronic Design Automation **2** (2008), 135-253.
15. I. V. Solís, M. Torres-Cisneros, J. G. Aviña-Cervantes, O. G. Ibarra-Manzano, E. Aguilera-Gómez, H. Plascencia-Mora, and J. J. Sanchez-Mondragón, Digital-Optical

Experimental Set up for in vitro Cell Tracking Based on Cross Correlation Technique,  
Frontiers in Optics, California, USA, 2009.

16. B. V. K. Vijaya Kumar, *Appl. Opt.* **31** (1992), 4773-4801.
17. A. Alfalou, G. Keryer, and J. de Bougrenet de la Tocnaye, *Appl. Opt.* **38** (1999), 6129-6135.
18. J. L. Horner and P. D. Gianino, *Appl. Opt.* **23** (1984), 812–816.
19. C. J. C. Burges, *Data Mining and Knowledge Discovery* **2** (1998), 121–167.
20. A. Alfalou, M. Elbouz, Ali Mansour, and G. Keryer, *J. Opt.* **12** (2010), **115403** (12 pp).
21. J. P. Egan, *Signal Detection Theory and ROC Analysis* (Academic Press Series in Cognition and Perception, New York, (1975).
22. Pointing Head Pose Image (PHPID),  
[http://www.ecse.rpi.edu/~cvrl/database/other\\_Face\\_Databases.htm](http://www.ecse.rpi.edu/~cvrl/database/other_Face_Databases.htm).
23. A. V. Oppenheim and J. S. Lim, *Proceeding of IEEE*, (1981), pp. 529–541.
24. A. Alfalou, M. Elbouz, M. Jridi, and A. LouSSERT, Vol., *Proceeding of SPIE* (2009) 7486.
25. J. L. Horner, *Appl. Opt.* **31** (1992), 165-166.
26. H. Zhou and T.- H. Chao, *Proceeding of SPIE* (1999) **3715**, 394.
27. S.H. Park, D.R. Shires and B.J. Henz. 3rd ed. DoD HPCMP Users Group Conference (2008). 366-370.
28. K. Moreland and E. Angel, *Proceeding of Graphics Hardware*, (2003), pp. 112-136.
29. V. Garcia, *Suivi d'objets d'intérêt dans une séquence d'images : des points saillants aux mesures statistiques*, Nice, 2008.

30. NVIDIA CUDA Compute Unified Device Architecture Programming Guide, V. 2.0, 2008.
31. M. Frigo, S.G. Johnson, Proceeding of IEEE, (2005), pp. 216-231.
32. See Supplementary Material at <http://www.youtube.com/watch?v=-Ca8apgsebY>

## **Table captions**

Table 1 : Influence of the pre-processing step on the correlator's robustness against rotation of the face. Symbols are : NP (no pre-processing); WP (with pre-processing).

Table 2: Experimental correlation results using a GPU GeForce 8400 GS and 2, 3 or 4 classes at the output of the classifier level.

					
NP	PCE=0.06	PCE=0.12	PCE=0.13	PCE=0.11	PCE=0.06
WP	PCE=0.13	PCE=0.22	PCE=0.51	PCE=0.24	PCE=0.18

Table 1 : Ouerhani *et al.*

	SPOF manufactured with 3 reference images			SPOF manufactured with 5 reference images		
	2	3	4	2	3	4
Number of output class of first level						
Mean recognition rate (%)	74	83	85	80	85	87
Maximum run time for identification (ms)	90	119	147	90	119	147

Table 2: Ouerhani *et al.*

## Figure Captions

Fig.1: Principle of the correlation algorithm developed in this study.

Fig.2: Database used: (a) 8 subject faces considered in this work; (b) several examples of rotations of the face of one subject.

Fig.3: Illustrating the algorithm used to process the target image: contour determination and content smoothing.

Fig.4: Influence of the number of iterations on the correlation plane.

Fig. 5: ROC curve with pre-processing.

Fig.6 : Correlation results: (a) target image; (b) reference image used to fabricate the correlation filter; (c) the correlation plane without the pre-processing; (d) the correlation plane obtained after 4 runs of our algorithm.

Fig.7: Influence of (a) weighting parameters and (b) pre-processing on the performances of the optimized segmented 3-reference filter.

Fig.8: Correlations results obtained with the database shown in Fig.2 and using a 5-reference OT-MACH filter.

Fig.9: Decomposition principle of the database acquisition into several classes.

Fig.10: Two-level based correlation algorithm: (a) classification of the target images; (b) identification of the target images.

Fig.11: Correlation results of the entire database using 8 class filters, i.e. PCE versus face position.

Fig.12: Second level correlation of face identification: experimental results of the PCE versus the face position.

Fig.13: A comparison of CPU and GPU performances.

Fig.14: GPU architecture.

Fig.15: CUDA architecture [30].

Fig.16: (a) Run time (in milliseconds) comparing GPU (NVIDIA GeForce 8400 GS) vs CPU (Pentium Dual Core CPU 2.50 GHz, and Intel-Core 2 CPU 2.00 GHz) of the numerical correlation algorithm for 1000 filters. (b) Same as in (a) for 15 filters.

Fig.17: Influence of the size of the image on the run time for the four architectures considered in Fig. 16.

Fig.18: Different positions and angles used to fabricate the SPOF.  $D$  is the depth distance, and  $\theta$  denotes the angle of the rotation of the face.

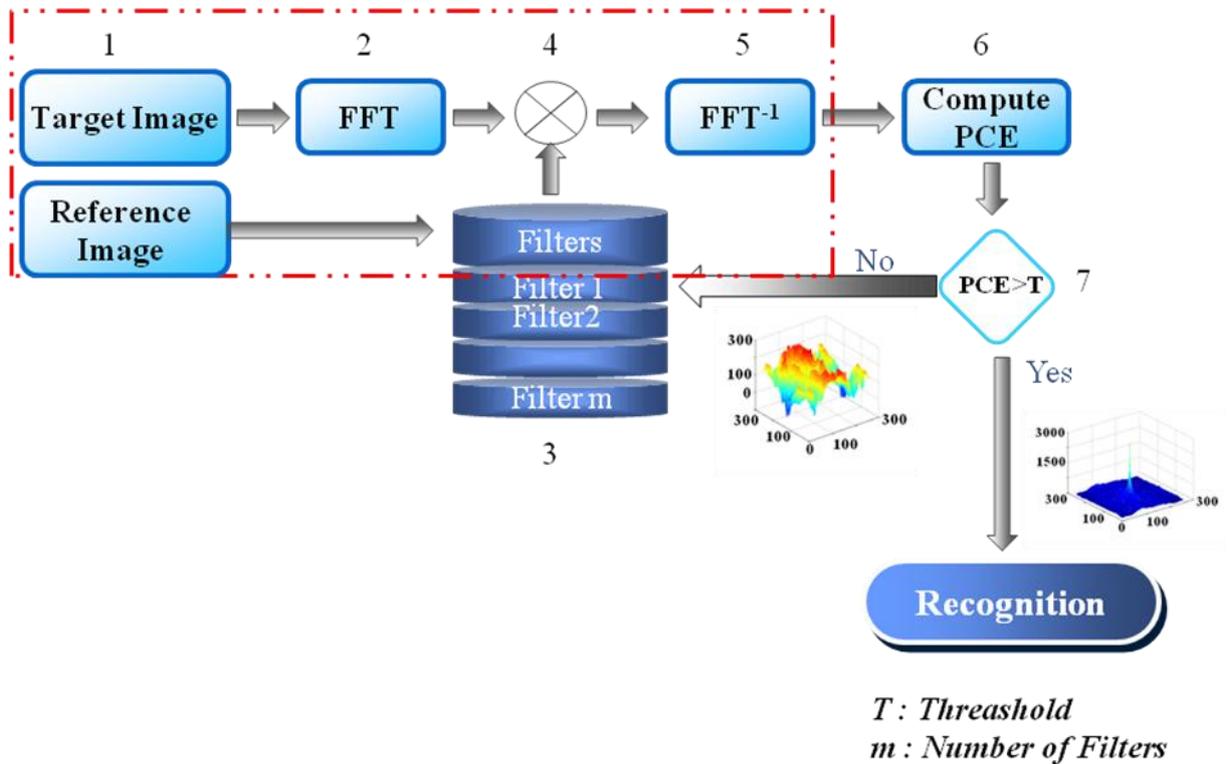


Fig.1: Ouerhani *et al.*

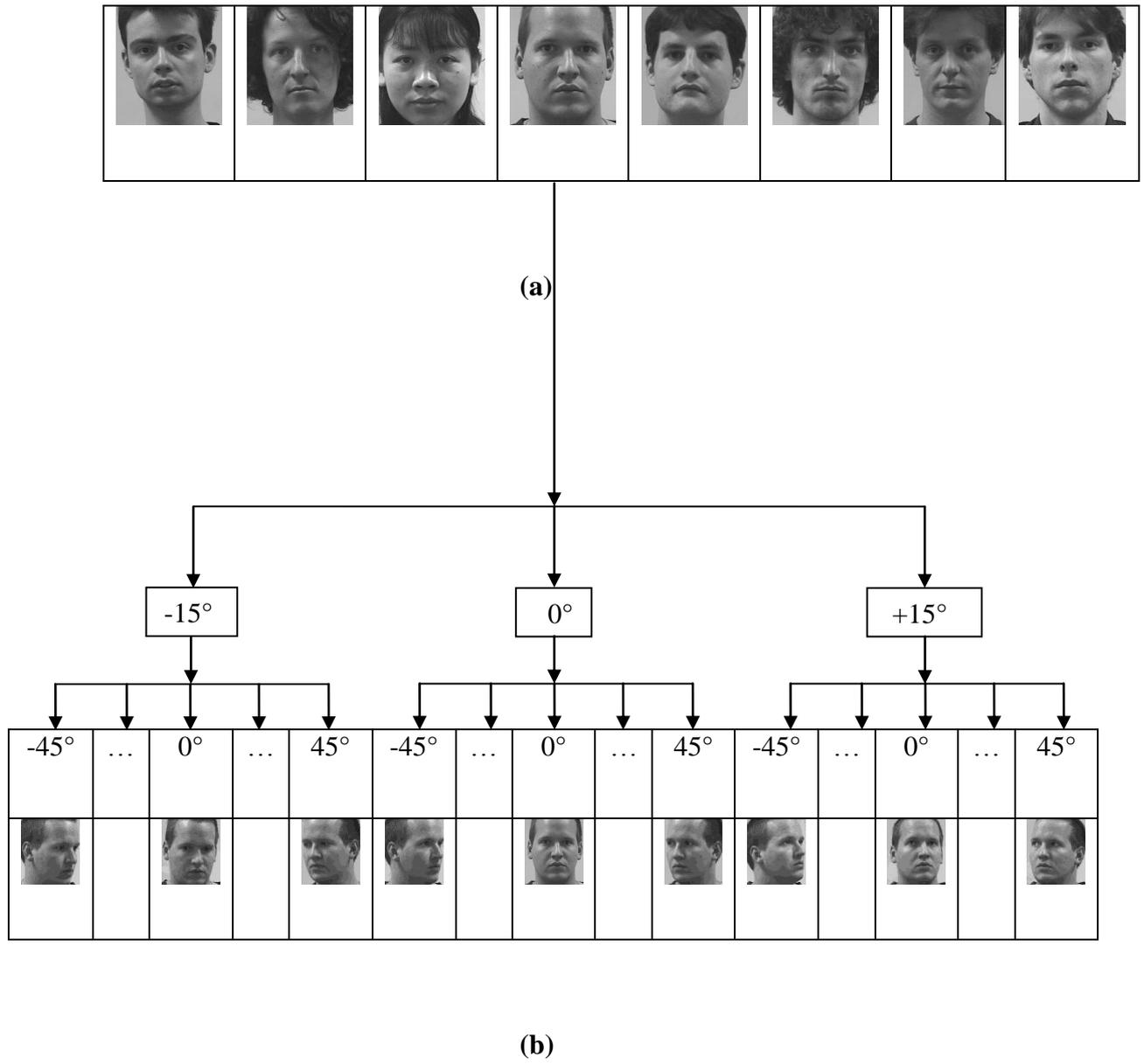


Fig.2: Ouerhani *et al.*

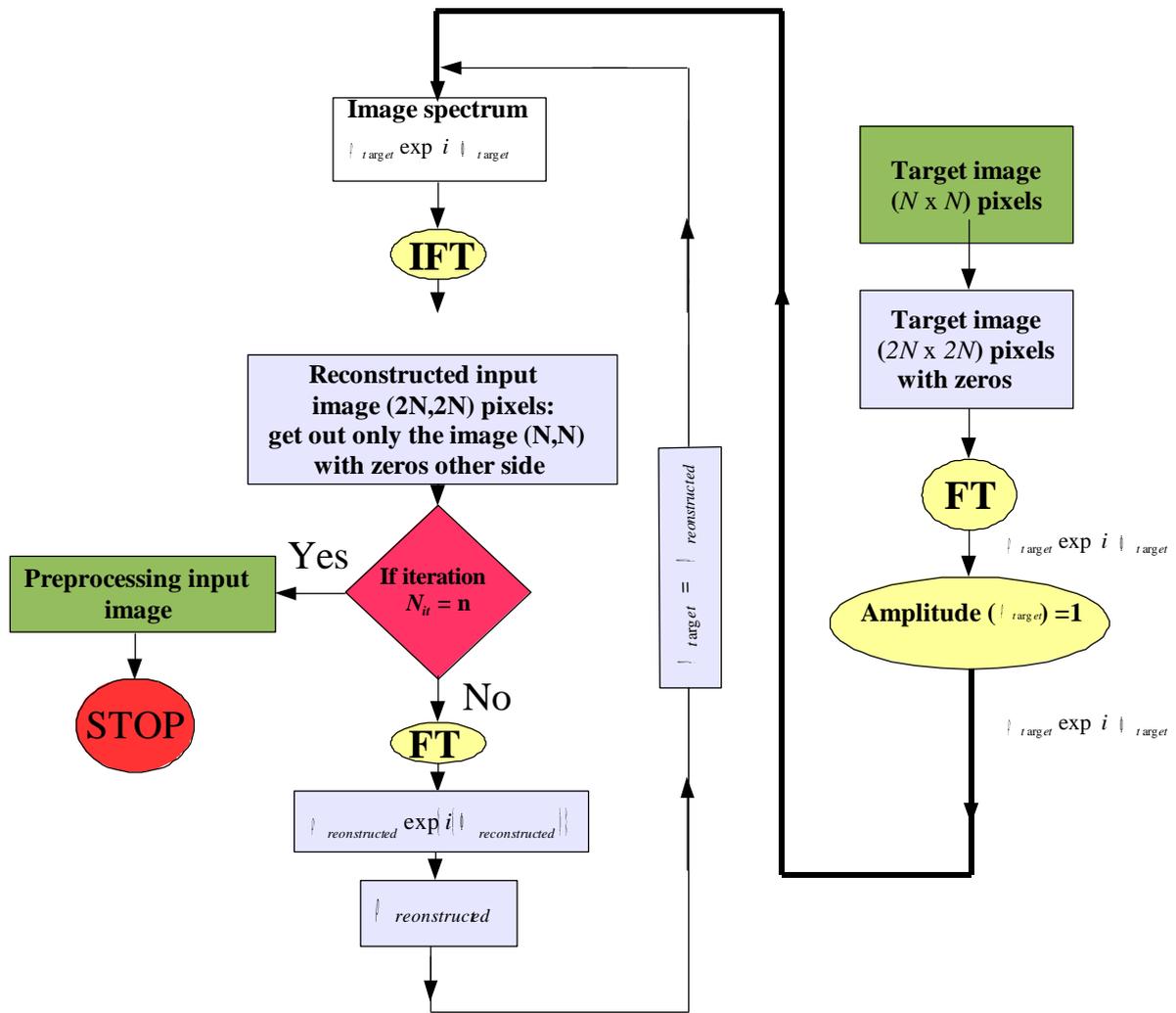


Fig.3: Ouerhani *et al.*

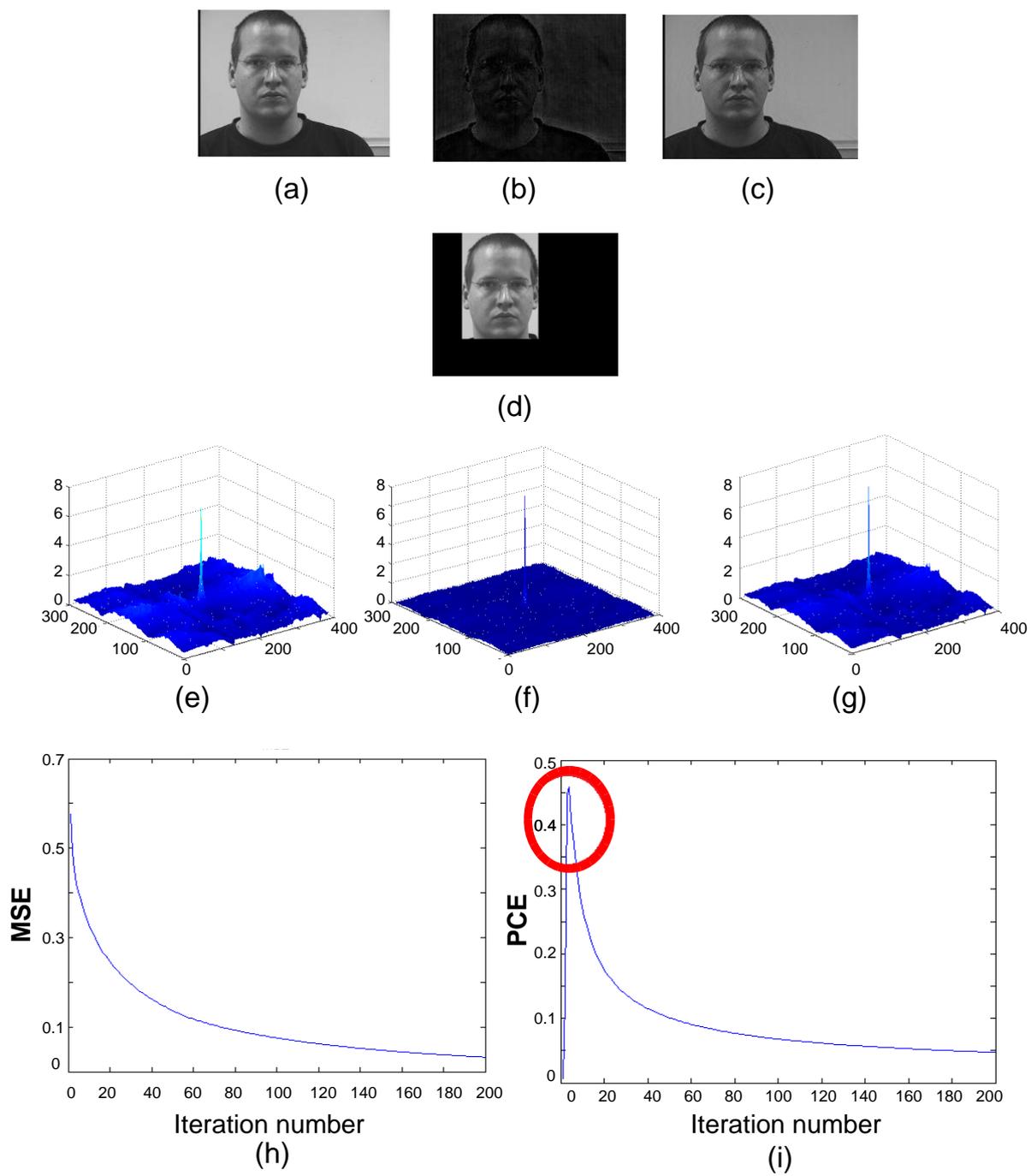


Fig.4 : Ouerhani *et al.*

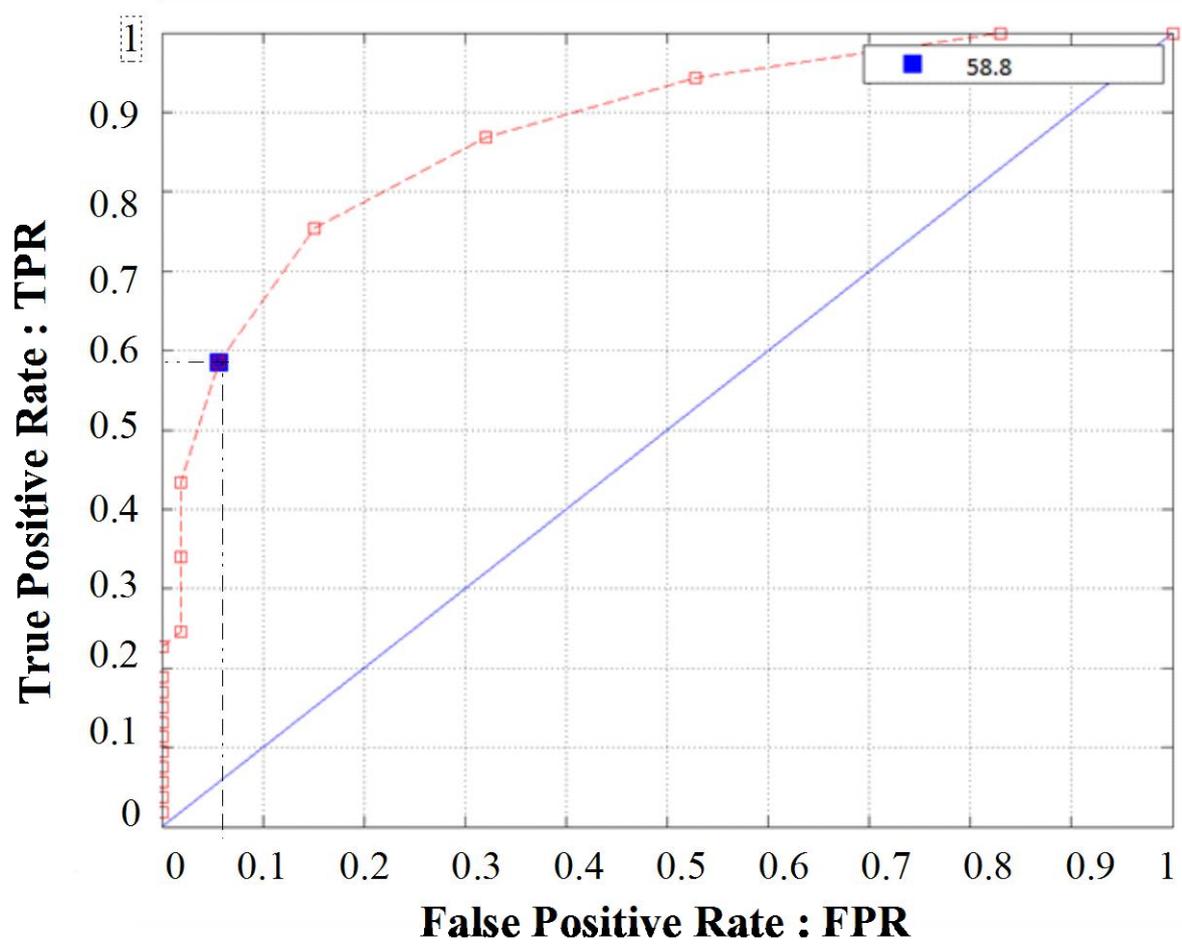


Fig. 5 : Ouerhani *et al.*

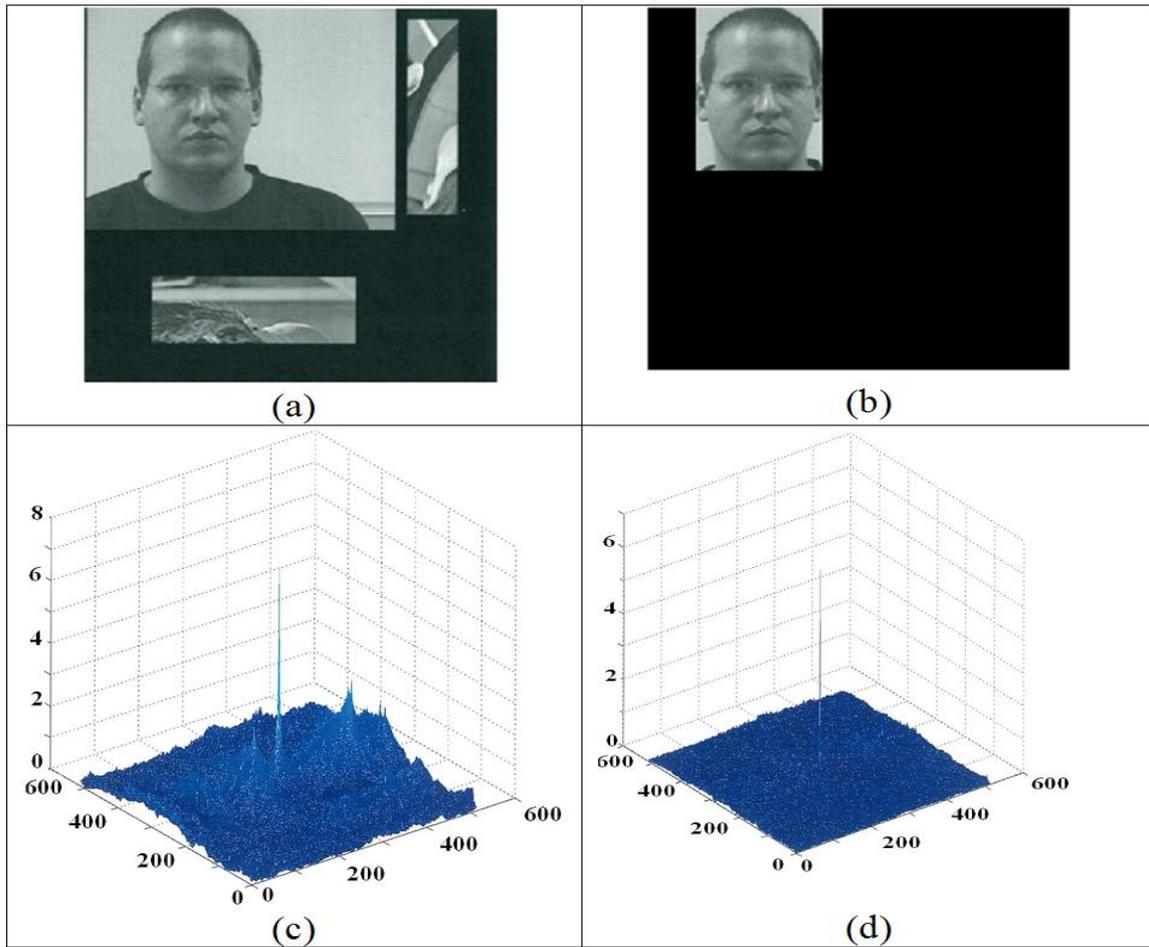


Fig.6: Ouerhani *et al.*

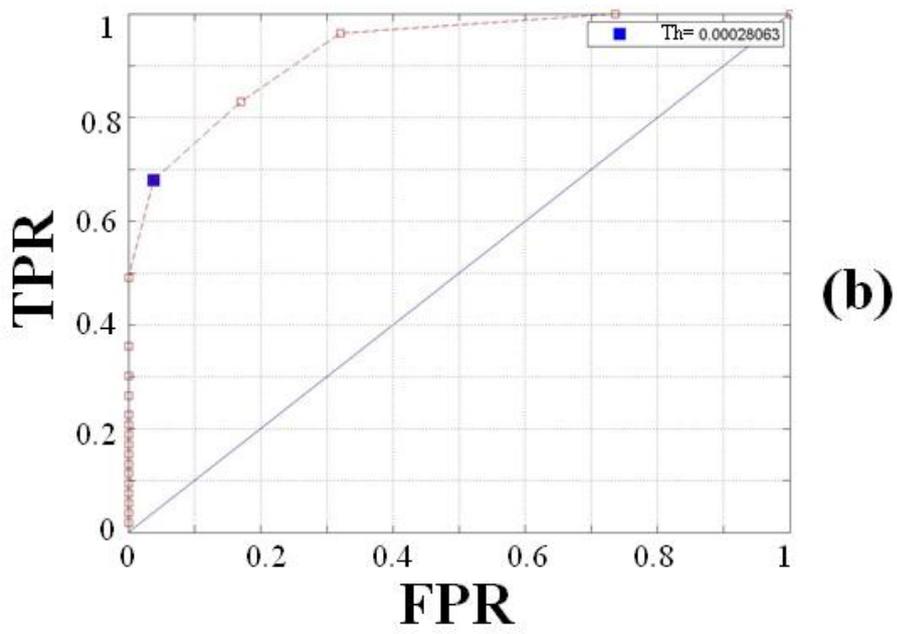
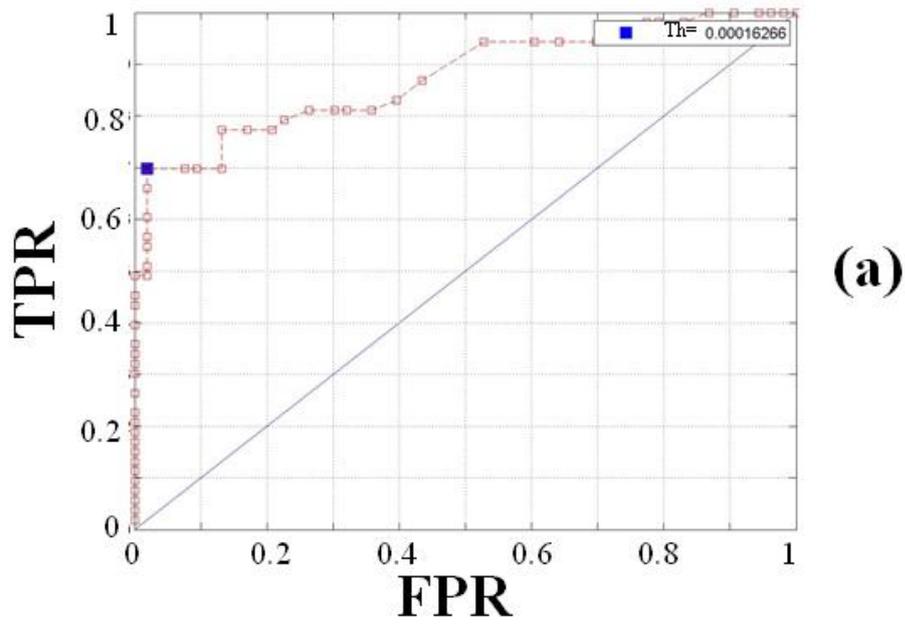


Fig.7: Ouerhani *et al*

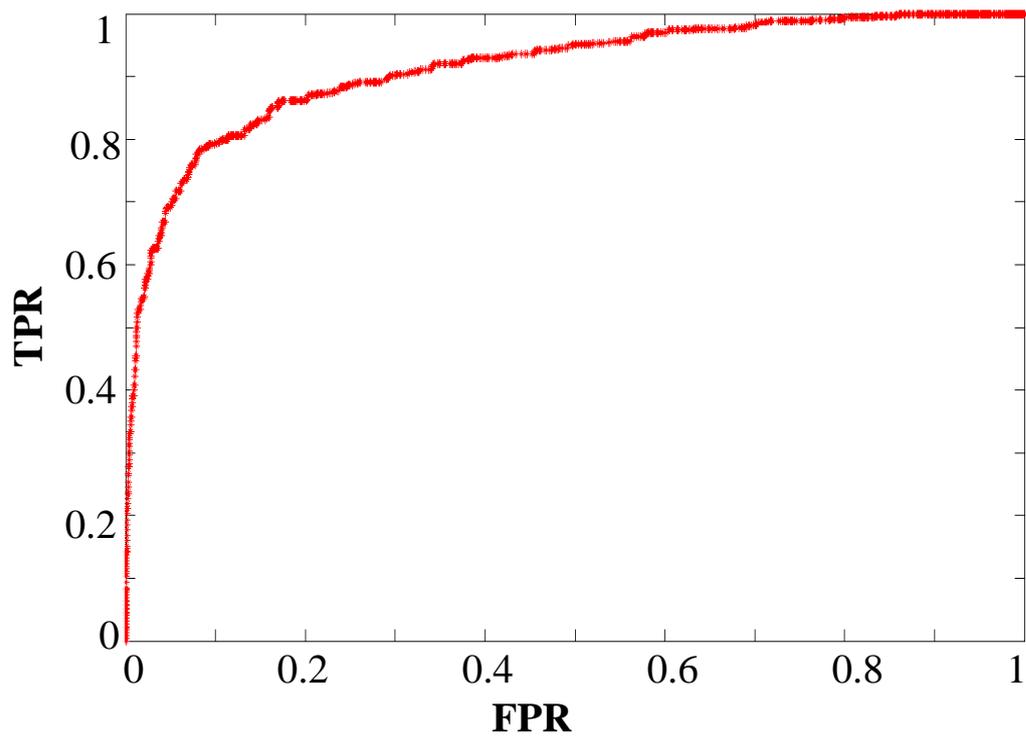
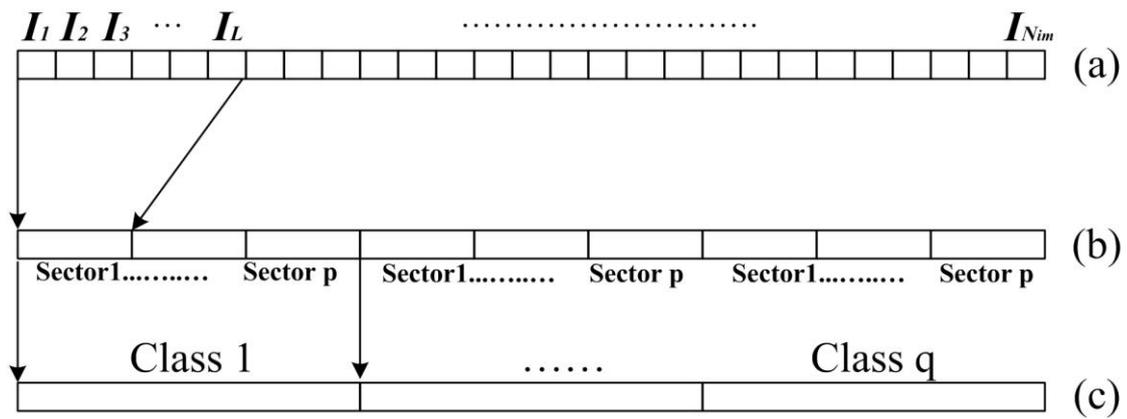
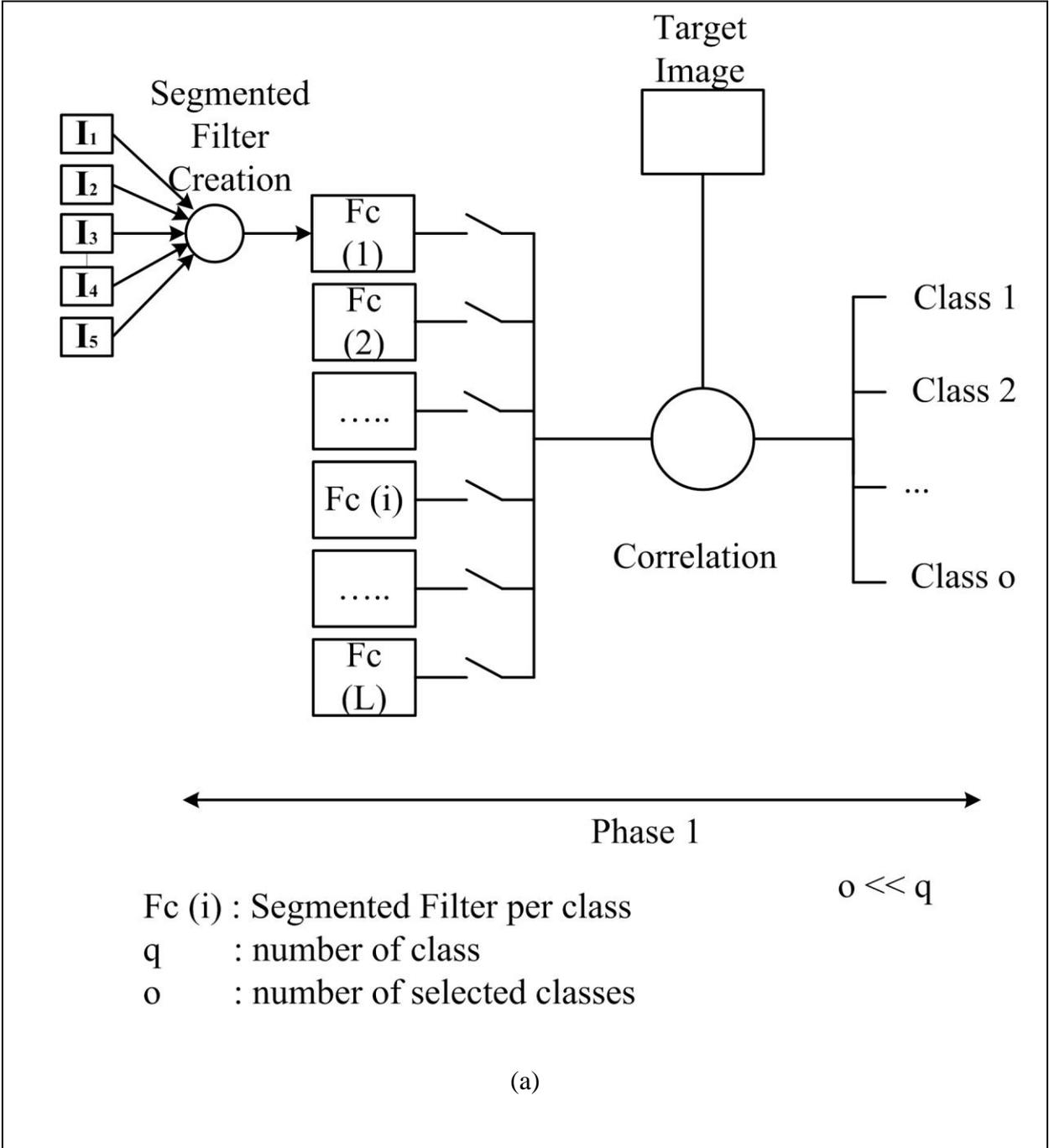


Fig.8: Ouerhani *et al*



$N_{im}$  : number of images  
 $p$  : number of sectors per class  
 $q$  : number of class  
 $L$  : number of images per sector

Fig.9: Ouerhani *et al*



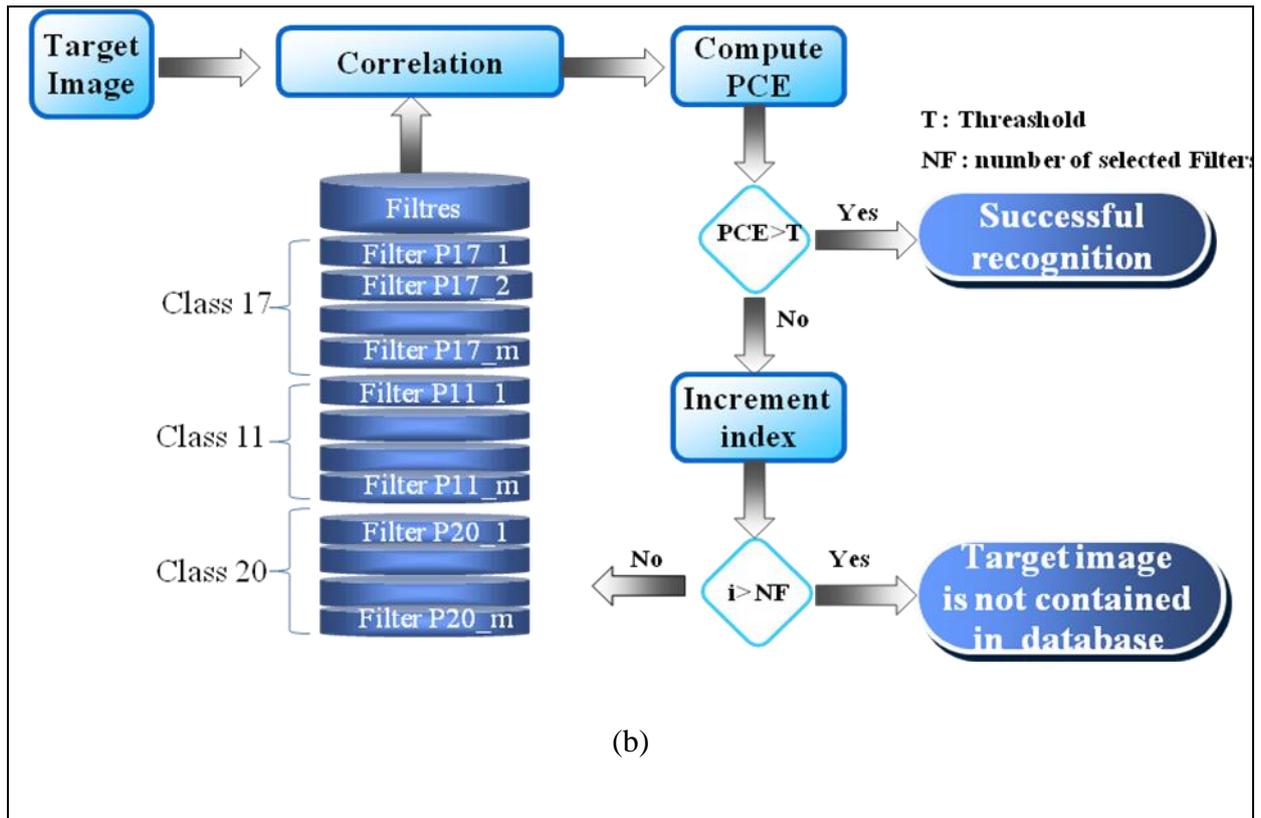


Fig.10: Ouerhani *et al.*

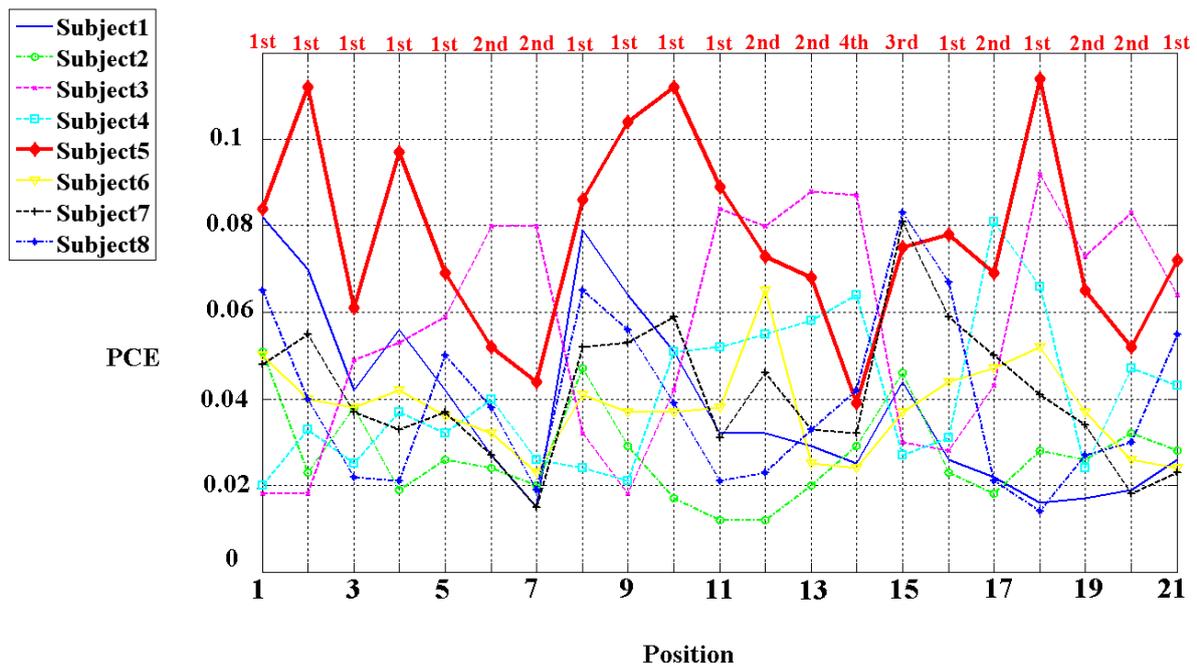


Fig.11: Ouerhani *et al.*

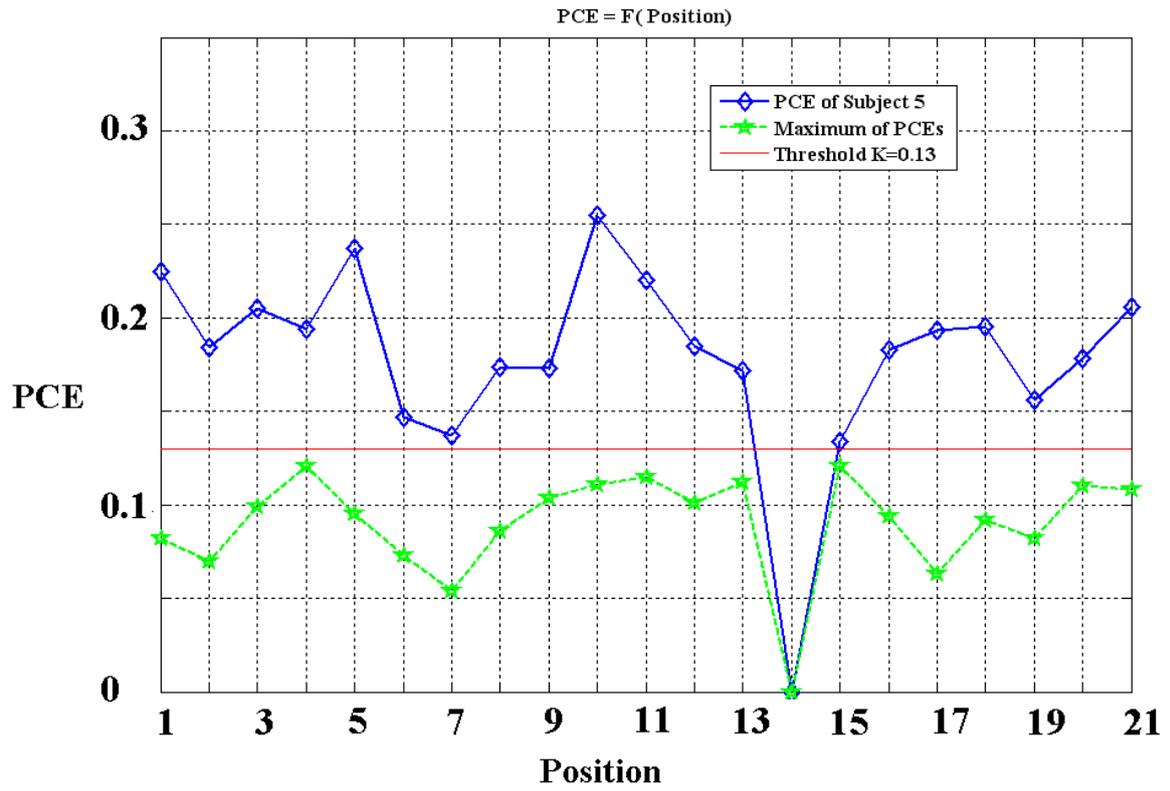


Fig.12: Ouerhani *et al.*

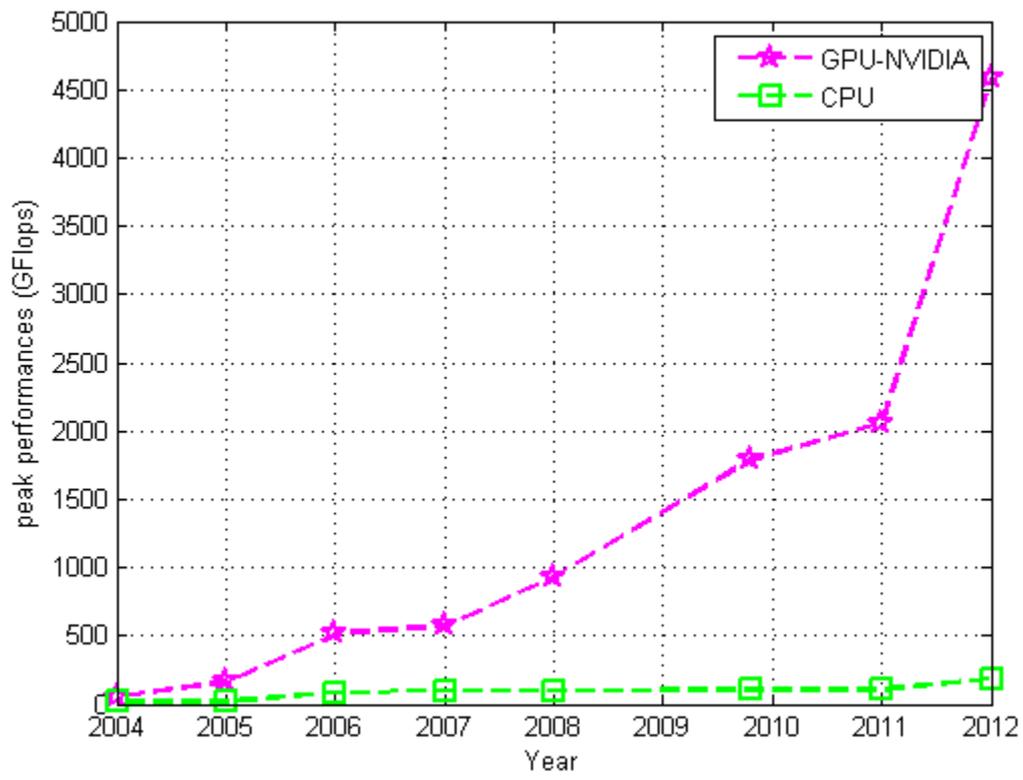
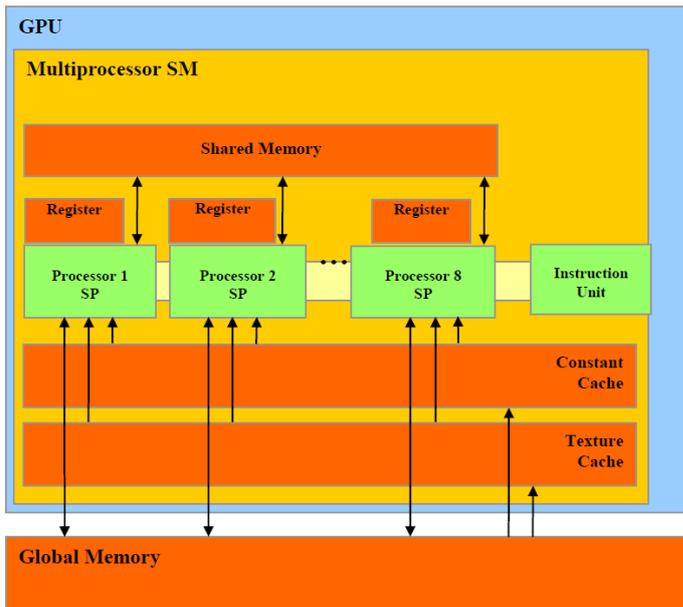
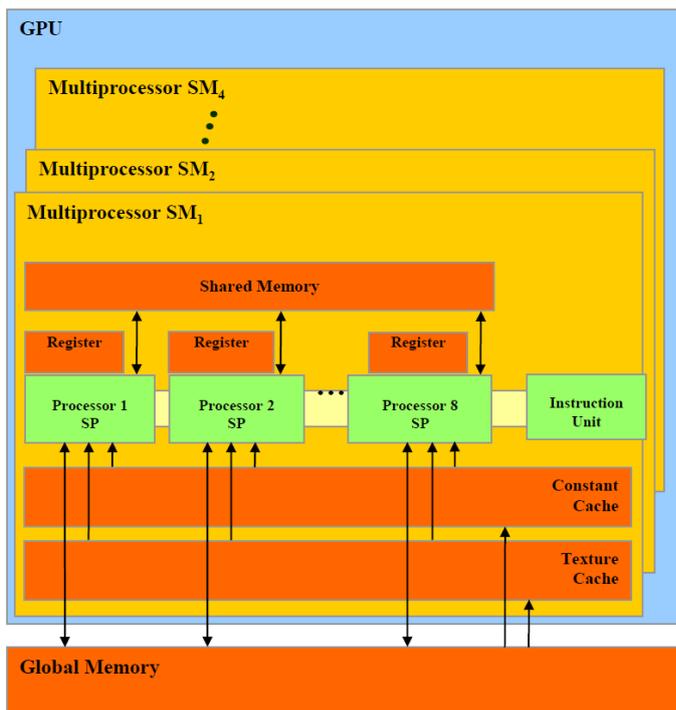


Fig.13: Ouerhani *et al.*



(a)



(b)

Fig.14: Ouerhani *et al.*

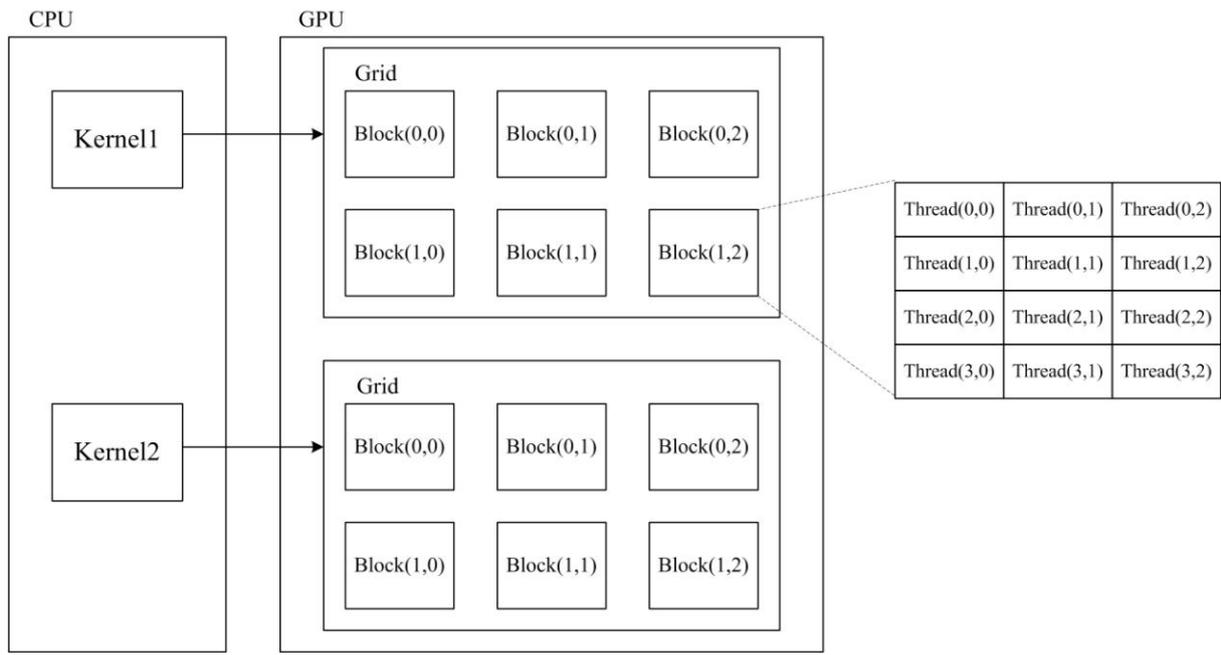


Fig.15: Ouerhani *et al.*

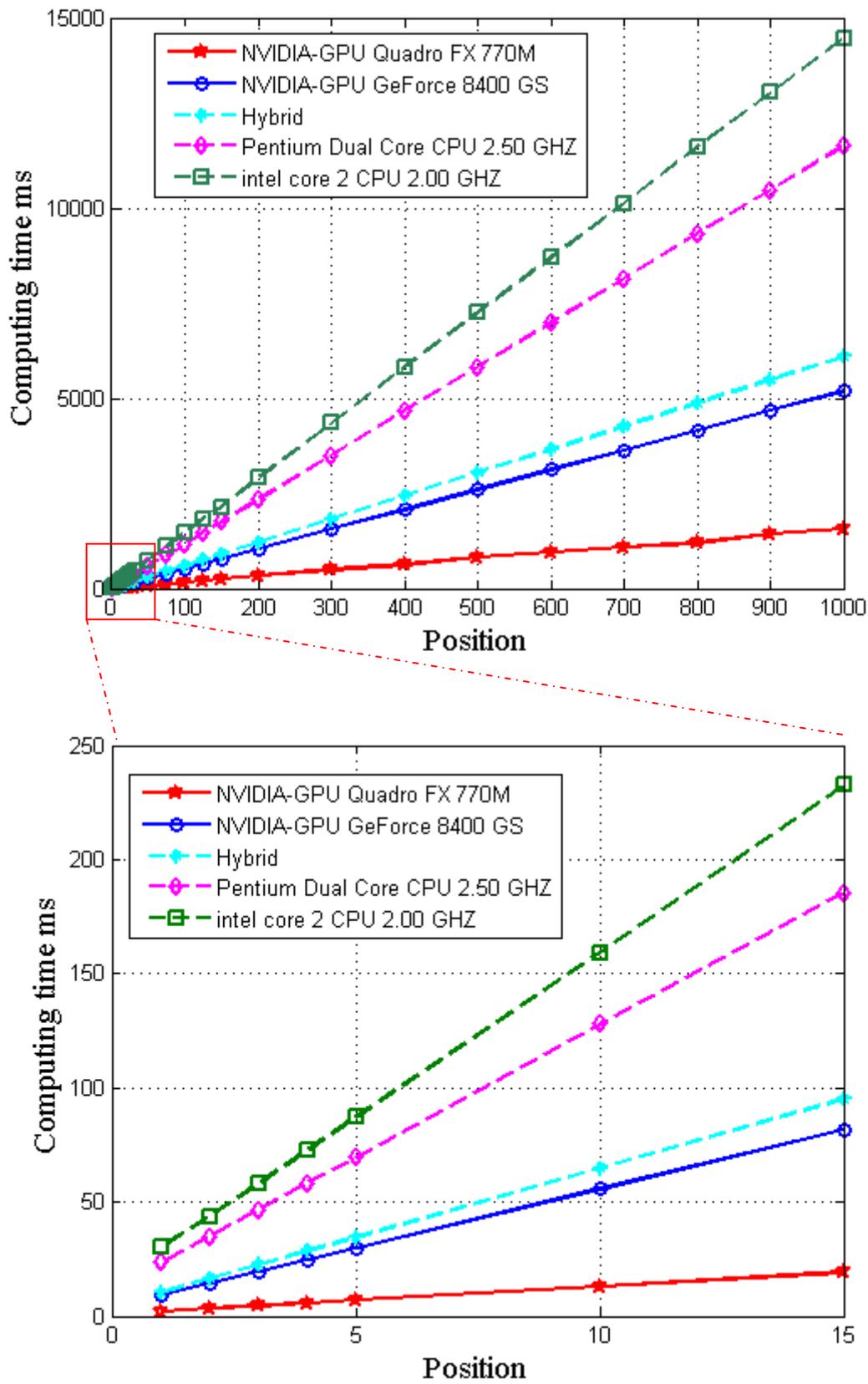


Fig.16: Ouerhani *et al.*

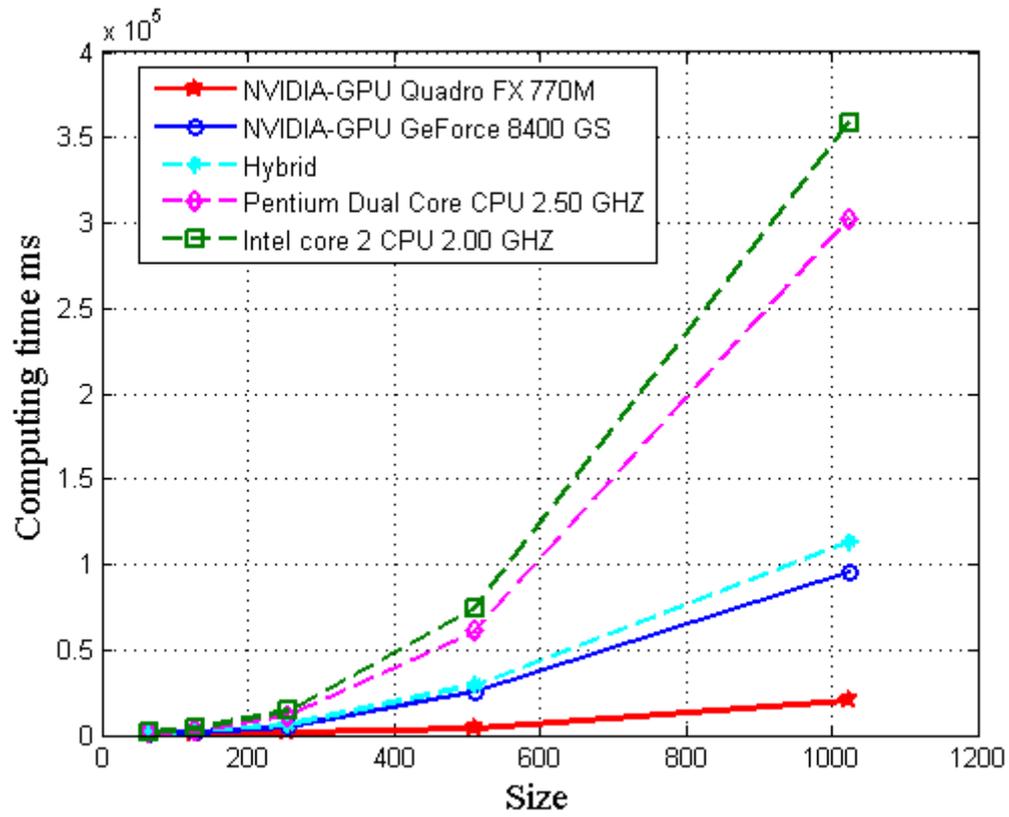


Fig.17: Ouerhani *et al.*

$D=1\text{ m}$	$D=1\text{ m}$	$D=1\text{ m}$	$D=60\text{ cm}$	$D=60\text{ cm}$	$D=60\text{ cm}$	$D=30\text{ cm}$	$D=30\text{ cm}$	$D=30\text{ cm}$
$\theta = -30^\circ$	$\theta = 0^\circ$	$\theta = +30^\circ$	$\theta = -30^\circ$	$\theta = 0^\circ$	$\theta = +30^\circ$	$\theta = -30^\circ$	$\theta = 0^\circ$	$\theta = +15^\circ$
								

Fig.18: Ouerhani *et al.*