

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

MOUNGLA Mohamed

&

DAOUD Rabah

pour l'obtention du diplôme de Master en électronique option Système de Vision et Robotique

Thème

Reconnaissance de face humaine en temps réel via Webcam avec la méthode des faces propres

Proposé par : Mr .NAMANE Abderrahmane

Année Universitaire 2011-2012

Remerciements

C'est pour nous un plaisir autant qu'un devoir de remercier toutes les personnes qui ont pu contribuer de près ou de loin à l'établissement de ce projet, qui m'ont aidé, m'ont soutenu et ont fait sorte que ce travail ait eu lieu.

Ainsi, j'exprime ma gratitude et je tiens à remercier monsieur Abderrahmane Namane, qui m'ont encadré, et qui n'ont épargné aucun effort pour m'orienter afin que je puisse mener à bien ce projet.

Je remercie également les membres de jury.

Je saisis aussi l'occasion pour remercier tout le corps professoral et administratif de la faculté des sciences et technologie de Blida, en particulier les membres de l'SVR.

Enfin, je remercie affectueusement mes parents, qui ont toujours su me faire confiance et me soutenir sans compter dans mes études.

ملخص:

التعرف على الوجه هي تكنولوجيا حيوية في رواج، و تستخدم على نطاق واسع في تطبيقات المراقبة. في الأدب، هناك عدة طرق عالمية، محلية وهجينة للتعرف على الوجه. والهدف من هذا المشروع هو التقديم وشرح خوارزمية الوجه الخاص، وتقديم نتائج جميع الاختبارات من هذه الخوارزمية التي قمنا بتنفيذها.

كلمات المفاتيح: وجه الخاص، التعرف على الوجه، الإضاءة

Résumé :

La reconnaissance de visage est une technologie biométrique en vogue, elle est très utilisée dans les applications de contrôle d'accès. Dans la littérature, on trouve plusieurs méthodes globales, locales et hybrides de reconnaissance de visage. Le but de ce projet est d'introduire et d'expliquer l'algorithme d'Eigenface, et présenter le résultat de l'ensemble des tests de cet algorithme qu'on a implémenté.

Mots clés : visage propre ; reconnaissance de visage ; éclairage.

Abstract :

The face recognition is a popular biometric technology, which is highly used in access control applications. In the literature, we find several holistic, hybrid and feature-based methods of face recognition. The goal of this project is introduce and to explain the Eigen Faces algorithms, and especially to present all the results of the tests of the set algorithm which we have implemented.

Keywords: biometrics; face recognition; illumination.

Listes des acronymes et abréviations

RGB/RVB: Red, Green, Blue/Rouge, Vert, Bleu.

OpenCV: Open Source Computer Vision.

VAO : Vision Assistée par Ordinateur.

SVM : Support Vector Machine.

CCD : Charge-Coupled Device.

ACP/PCA : Analyse Composant Principale /Principal Component Analyse.

RAM: Random Access Memory.

IPP: Integrated Performance Primitives.

I/O: Input/Output.

DPI : Dot Per Inch.

Table des matières

Introduction générale	1
CHAPITRE 1 : VISION PAR ORDINATEUR.....	3
1.1 Introduction	3
1.2 Image numérique	4
1.3 Type d'image.....	6
1.4 Traitement d'image	8
1.5 Conclusion	11
CHAPITRE 2 : LA RECONNAISSANCE ET AU SUIVI FACIALE	12
2.1 Introduction	12
2.2 Image numérique	13
2.3 Prétraitement	14
2.4 Extraction de paramètres	18
2.5 Base d'apprentissage	23
2.6 Classification	24
2.7 Test et résultat	24
2.8 Conclusion	25
CHAPITRE 3 : IMPLEMENTATION, TESTS ET RESULTATS	26
3.1 Introduction	26
3.2 Mise en situation	26
3.3 La librairie OpenCV	28
3.4 Implémentation	28
3.5 Interfaces de l'application	31

3.6 Test	37
3.7 Conclusion	42
Conclusion Générale	44
Annexe	46
Bibliographie	51

Liste des figures

Figure 1.1. Une image et son histogramme.	9
Figure 1.2. Image à niveau de gris	10
Figure 1.3. Image histogramme égalisé.....	10
Figure 1.4. Histogramme d'une image à niveaux de gris	10
Figure 1.5. Histogramme d'une image égalisé	10
Figure 2.1. Principe général d'un système de reconnaissance de visages	13
Figure 2.2. Exemple des types de caractéristiques utilisées par Viola et Jones	15
Figure 2.3. L'image intégrale.....	16
Figure 2.4. Exemple de calcul d'une caractéristique Haar	17
Figure 2.5. Cascade de classifieurs AdaBoost	18
Figure 2.6. Exemple d'eigenfaces générés avec 60 images de références [15]	
Figure 2.7. Exemple de quelque visage dans la base de données.....	23
Figure 3.1. Menu <i>de mot de passe</i>	31
Figure 3.2. Menu <i>modification de mot de passe</i>	31
Figure 3.3. Menu <i>principal</i>	32
Figure 3.4. Menu <i>de reconnaissance en temps réel</i>	33
Figure 3.5. <i>Récupération de la vidéo</i>	33
Figure 3.6. <i>Résultat de détection et reconnaissance humain en temps réel</i>	34
Figure 3.7. <i>Résultat de pointage automatique</i>	34
Figure 3.8. Détection et reconnaissance de visage	35
Figure 3.9. Résultat sur image fixe	35
Figure 3.10. Base de données.....	36
Figure 3.11. Fiche « A propos »	36
Figure 3.12. Représentation de test situation.....	37
Figure 3.13. Quelques exemples de la base de tests	38
Figure 3.14. Quelques exemples de la reconnaissance avec un seuil = 0.9	39
Figure 3.15. Quelques exemples de la reconnaissance avec un seuil = 0.5.....	39
Figure 3.16. Résultat de pointage sur image fixe	40

Figure 3.17. Performance de visages propres	41
Figure 3.18. Résultat de détection et reconnaissance de visage en temps réel	41
Figure 3.19. Résultat de pointage automatique	42

Liste des tableaux

Tableau 1.1. Les niveaux de gris.....	7
Tableau 1.2. Les couleurs RVB.....	8
Tableau 3.1. La différence entre les langages des programmations.....	29
Tableau 3.2. Tests de la méthode des faces propres avec différent seuils.....	40

Introduction générale

De nos jours, on parle de plus en plus de l'insécurité dans divers secteurs ainsi que des moyens informatiques à mettre en œuvre, pour contrer cette tendance : le contrôle d'accès aux ordinateurs, l'e-commerce, les opérations bancaires basées sur l'identification du demandeur, etc.

Les techniques appliquant la biométrie actuellement visent à établir l'identité d'une personne en mesurant une de ses caractéristiques propres. Ces dernières sont variables, les unes plus fiables que d'autres, mais toutes doivent être infalsifiables et uniques pour pouvoir être représentatives d'un seul et même individu. D'autre part, comme nous allons le voir, les caractéristiques physiques sont loin d'être aussi parfaites et précises, qu'on pourrait le penser et l'on atteint très vite des limites pour ces techniques.

L'intérêt des applications utilisant la biométrie se résume en trois classes : faciliter le mode de vie, éviter la fraude et contrôler la population.

L'objectif du projet est d'étudier les propriétés de la méthode de visages propres.

On développera un système capable :

- d'identifier un visage parmi une base de données de visages de référence
- de déterminer si la scène vidéo contient un visage présent dans la base de données
- de reconnaître si une image représente un visage ou non.

Ce mémoire est organisé en trois chapitres : Dans le premier chapitre on présente des généralités sur la vision artificielle. En deuxième chapitre on expose l'état de l'art des méthodes de détection et reconnaissance faciale. Le dernier

chapitre a été réservé à l'étude puis l'implémentation d'algorithme choisis sous c++
Builder, en utilisant la librairie OpenCV.

Chapitre 1 Vision par ordinateur

1.1 Introduction

« Voir » c'est, entre autres, discerner et reconnaître les formes, les couleurs et les textures du monde qui nous entoure.

On pourrait croire, à tort, que nous avons besoin uniquement de nos yeux pour cela, mais la réalité est tout autre : nos yeux ne servent qu'à capter le signal contenu dans les rayons lumineux qui viennent frapper nos rétines, pas à en extraire des informations. Ça, c'est le travail de la zone de notre cerveau que nous appelons le cortex visuel, cela veut dire que la vision, c'est l'association entre le sens de la vue et un ensemble de processus cognitifs pour lesquels nous avons besoin d'utiliser notre cerveau.

Aujourd'hui, cela fait déjà bien longtemps que nous avons donné le sens de la vue aux ordinateurs grâce aux caméras numériques, mais si l'on se fie à la définition que nous venons de formuler, cela ne suffit pas à les doter de vision : il leur manque encore la capacité à extraire des informations des images et des flux vidéo, de manière à percevoir, analyser et comprendre le monde qu'ils observent.

C'est ici qu'intervient ce vaste domaine de recherche qu'est la vision par ordinateur, à la croisée des chemins entre les mathématiques, le traitement du signal et l'intelligence artificielle.

La vision par ordinateur se base essentiellement sur le traitement d'image pour effectuer des tâches comme la segmentation d'objets, la reconnaissance de formes, la détection du mouvement, le suivi de cibles et autres techniques. Pour pouvoir effectuer ces tâches, on utilise des outils qu'on appelle : outils de traitement d'images, donc on parlera de quelques notions de base de traitement d'image, plus précisément les outils qu'on utilise au cours de notre projet (la détection de cibles), mais avant on présente ce qu'est l'image numérique.

1.2 Image numérique

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques) [4].

La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeurs dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts). C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur [5].

On distingue ainsi deux types d'image numérique : l'image en pixels et l'image vectorielle.

L'image en pixels se présente sous forme d'une grille constituée d'une multitude de points, comparable à une mosaïque : c'est ce qu'on appelle les pixels. L'intérêt de ces images est de reproduire des gradations subtiles de couleurs. L'image vectorielle est issue de formules géométriques, c'est-à-dire un ensemble de points reliés par des lignes et des courbes formant un tracé. Il est possible d'agrandir une image vectorielle sans perdre de la qualité initiale [4].

1.2.1 Pixel

Mot anglais issu de la contraction et de la juxtaposition de "Picture" et de "élément". Un pixel correspond au plus petit élément d'une surface d'affichage auquel on puisse associer individuellement une couleur (ou un niveau de gris) et une intensité. Dans le cas d'un écran monochrome, le pixel s'identifie à un point et, dans le cas d'un écran couleur, il est constitué de trois points de couleurs différentes (rouge, vert, bleu). En variant l'intensité de chacun des points, on peut faire apparaître des milliers de couleurs différentes. La quantité de pixels composant l'écran détermine la résolution [6].

1.2.2 Résolution

On l'exprime en dpi (dot per inch) ou ppp (point par pouce). Elle indique le nombre de points distincts que le matériel peut inscrire sur le papier dans une distance d'un pouce, soit 2,54 cm. Plus ce nombre sera élevé, plus les détails de l'image seront finement rendus. La résolution est souvent notée comme le produit de deux chiffres, par exemple 4800x1440 : dans ce cas, on exprime la densité de points sur une surface d'un pouce carré, soit 6,45 cm². Dans l'exemple ci-dessus, l'imprimante sera capable de placer près de 7 millions de points élémentaires sur une surface de 6,45 cm². Evidemment, plus ce nombre de points sera élevé, plus la définition de l'image sera précis [6].

1.2.3 Taille d'une image

Pour connaître la taille d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre des cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. La taille de l'image est alors le nombre des pixels que multiplie la taille (en octet) de chacun de ces éléments [7].

Exemple : pour une image de 240 X 420 en Truecouleur :

Nombre de pixels :

$$240 \times 420 = 100800$$

Taille de chaque pixel : 24 bits / 8 = 3 octets

Le poids de l'image est ainsi égal à :

$$100800 \times 3 = 302.400 \text{ égal } 302.400/1024 = 295 \text{ Ko.}$$

1.2.4 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparent de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites (bruit).

1.2.5 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport : [5]

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

1.3 Type d'image

1.3.1 Image a niveau de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255.

Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux (en général on sauvegarde les images à 256 teintes de gris) [8]

Pour les images couleurs, un pixel dispose généralement de trois composants RVB. Un pixel gris a ses trois valeurs RGB identiques. Une méthode simple pour convertir une image couleur en niveau de gris pourrait être de calculer la moyenne des trois composantes RGB et d'utiliser cette valeur moyenne pour chacune des composantes :

$$Gris = \frac{(Rouge + vert + Bleu)}{3}$$

Couleur	BLANC	GRIS 12.5%	GRIS 25%	GRIS 37.5%	GRIS 50%	GRIS 62.5%	GRIS 75%	GRIS 87.5%	NOIR
Niveau de gris	255	224	192	160	128	96	64	32	0

Tableau 1.1. Les niveaux de gris.

1.3.2 Image binaire

Parmi les images numériques, les images binaires sont les plus simples. Bichromies (la plupart du temps noire et blanche) elles sont ontologiquement numériques c'est-à-dire que leur codage et leur décodage peuvent être faits directement vers la base 2. [8]

1.3.3 Image RVB

Image RVB, représente directement les valeurs de couleurs, au lieu d'être exprimée à travers une palette de couleurs ou colormap. Une image truecouleur est un tableau de dimension $m \times n \times 3$, avec $m \times n$ les dimensions de l'image et 3 représente les composantes de couleurs R, V et B. Donc pour chaque pixel (x, y) de l'image, la couleur est représentée par le triplet $(x, y, 1:3)$ [9].

R	V	B	Couleur
0	0	0	Noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
128	128	128	Gris
255	255	255	Blanc

Tableau 1 .2. Les couleurs RVB

1.4 Traitement d'image

Le traitement d'images désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Il n'y a pas création d'informations, mais mise en évidence de l'information pertinente déjà présente.

1.4.1 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image.

Pour diminuer l'erreur de quantification ou pour comparer deux images obtenues sous des éclairages différents ou encore pour mesurer certaines propriétés de l'image, on modifie souvent l'histogramme correspondant.

L'histogramme permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée (figure I.2).

L'histogramme peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci[10].

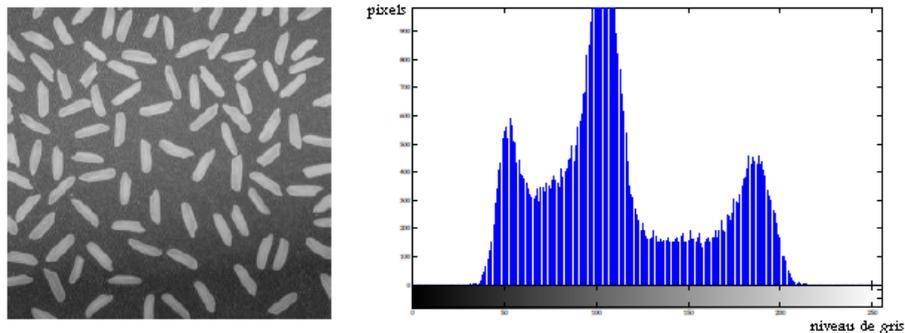


Figure 1. 1. Une image et son histogramme.

1.4.2 Egalisation d'histogramme

L'histogramme cumulé représente la distribution cumulée des intensités des pixels d'une image, c'est-à-dire le nombre de pixels ayant au moins une intensité lumineuse donnée.



Figure 1.2. Image à niveau de gris



Figure 1.3. Image histogramme égalisé

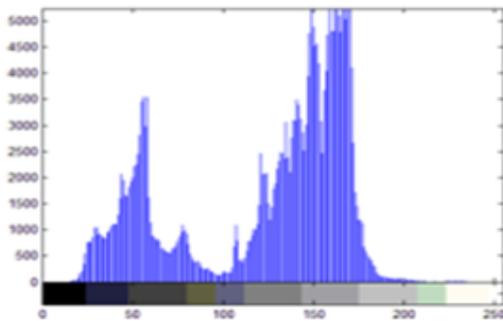


Figure 1.4. Histogramme d'une l'image à niveaux de gris.

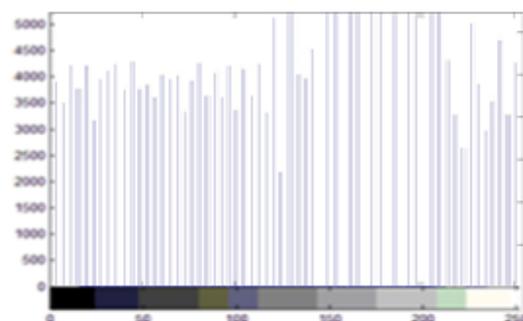


Figure 1.5. Histogramme d'une l'image Egalisé.

1.5 Conclusion

On a vu dans ce chapitre les généralités sur la vision artificielle, ainsi que les différentes notions de base du traitement d'image, on détaille en chapitre suivant une des techniques de la détection et reconnaissance faciale. On fait un état de l'art des dernières méthodes utilisées surtout pour la détection de visage et l'algorithme d'eigenfaces.

Chapitre 2 La reconnaissance et le suivi facial

2.1 Introduction

Dans la littérature, de nombreuses méthodes de détection et reconnaissance de visage ont été présentées ; une grande partie d'entre elles, peuvent être utilisées pour la détection de visage en temps réel. Malheureusement, la plupart de ces méthodes sont fondées sur un seul modèle de cible ou une seule modalité ; elles sont limitées à certains environnements spécifiques, typiquement pour la vidéo surveillance, des scènes statiques, contrôlés et connues à priori, sur lesquelles la méthode suit tous les objets dynamiques qui apparaissent. On peut admettre, sans aucun risque de se tromper, que jusqu'à aujourd'hui il n'existe pas une méthode suffisamment générale pour traiter avec succès et robustesse, la grande variété des cibles et conditions qui existent dans le monde réel. Dans ce chapitre on détaille la chaîne de reconnaissance et détection des visages en temps réel suivant l'organigramme présenté par Figure 2.1.

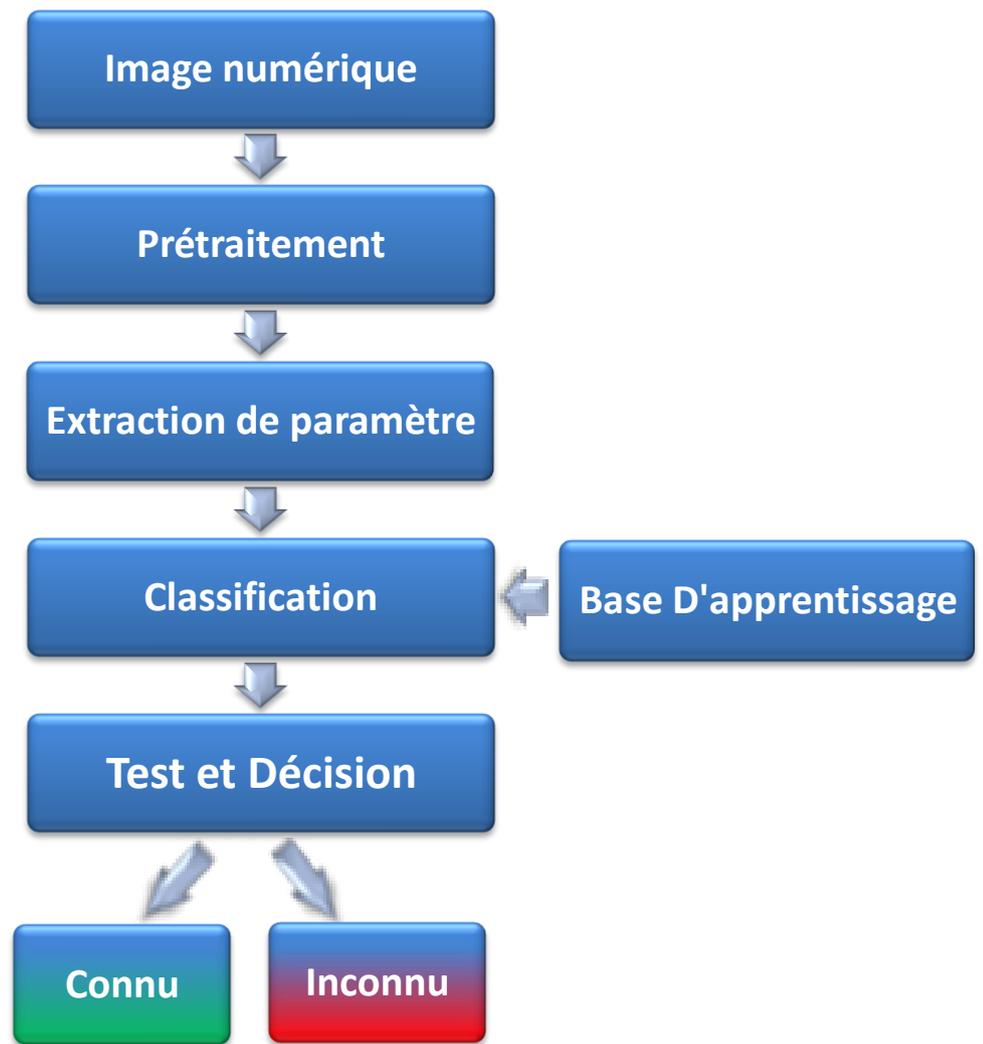


Figure 2.1. Principe général d'un système de reconnaissance de visages

2.2 Image numérique

Le traitement d'images est basé essentiellement sur l'image numérique qui est l'ensemble des pixels placés sous forme matricielle (on parle sur l'image numérique dans le premier chapitre).

2.3 Prétraitement

Cette étape est très importante avant de commencer le processus de reconnaissance du visage, et contient trois étapes fondamentales :

2.3.1 Conversion Image RVB à niveau de gris

Dans cette étape on convertit l'image codée sur trois plans RVB à une image codée sur un seul plan niveau de gris (Nous définissons l'image à niveau de gris dans le premier chapitre).

2.3.2 Égalisation

L'égalisation d'histogramme est une transformation des niveaux de gris dont le principe est d'équilibrer au mieux possible la distribution des pixels dans la dynamique gris (Nous définissons en détaille L'égalisation d'histogramme dans le premier chapitre).

2.3.3 Détection de visage

La détection du visage est actuellement implémentée dans la librairie OpenCV et repose sur des méthodes de classification des données (méthode de Haar) [22].

L'image est déclinée en différente taille, pour chacune des résolutions, une fenêtre de taille commune à chaque image est balayée de manière aléatoire (pour un gain de temps) [22].

Cette fenêtre va être ensuite utilisée pour découper une sous image et pour commencer l'analyse de visage humain. Par un processus d'apprentissage, un certain nombre de (boites) sont défini afin de sommer les pixels et définir une valeur type caractérisant le visage humain. Ces zones critiques ont été apprises automatiquement par des algorithmes en leur soumettant un grand nombre d'individus [22].

La méthode la plus connue est celle mise au point par Viola et Jones en 2001, permettant de détecter la présence de visage sur une image en temps réel [22].

a La méthode de Viola et Jones

Cette méthode propose une architecture pour combiner les classifieurs boostés en un processus en cascade, ce qui apporte un net gain en temps de détection.

- **Caractéristiques**

Les caractéristiques proposées par Viola et Jones, considèrent des fenêtres de détection (ou masques) délimitant des zones rectangulaires adjacentes ; les intensités de pixels de ces rectangles sont additionnées, formant des sommes dont la différence constitue une caractéristique. La figure ci-dessous donne des exemples de masques à 2, 3 ou 4 rectangles, dans lesquelles la somme de pixels délimités par la zone sombre est soustraite de la somme des pixels délimités par la zone claire. Ces masques ou fenêtres de détection, généralement de très petite taille au départ (typiquement 20×20 pixels), sont appliqués à toutes les positions de l'image, puis leurs tailles agrandies. Une caractéristique est donc un nombre réel qui code les variations du contenu pixellique à une position donnée dans la fenêtre de détection. La présence de contours ou un changement de texture sont ainsi traduits numériquement par les valeurs des Caractéristiques pseudo-Haar. Par exemple, un masque à 2 rectangles permet d'indiquer où se situe la frontière entre une région sombre et une région claire [22] (figure 2.2).

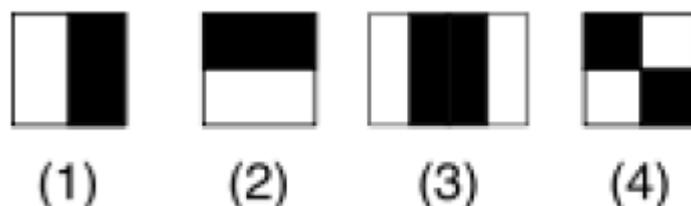
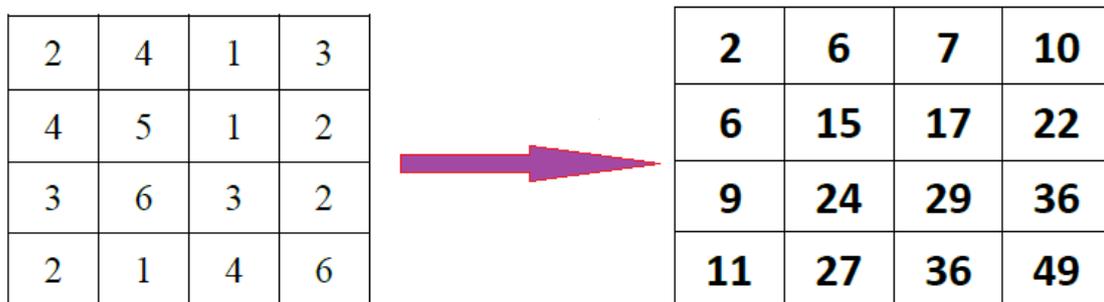


Figure 2.2. Exemple des types de caractéristiques utilisées par Viola et Jones

- **L'image intégrale**

A première vue, pour calculer chaque caractéristique, il semblerait nécessaire de parcourir l'ensemble des pixels de la zone, ce qui nécessite une puissance de calcul non négligeable. Pour remédier à ce problème, Viola et Jones ont introduit le concept d'image intégrale.

Une image intégrale est une matrice qui va contenir pour chaque pixel la somme de sa valeur et de l'ensemble des valeurs des pixels se trouvant au-dessus de lui et à sa gauche [22] (figure 2.3).



2	4	1	3
4	5	1	2
3	6	3	2
2	1	4	6

2	6	7	10
6	15	17	22
9	24	29	36
11	27	36	49

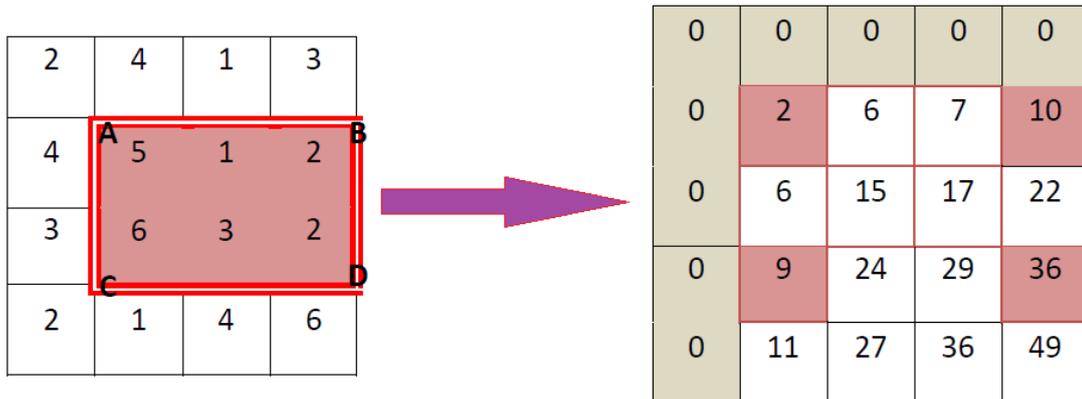
Figure 2.3 L'image intégrale

- **Calculs**

Les caractéristiques sont calculées à toutes les positions et à toutes les échelles dans une fenêtre de détection de petite taille, typiquement de 24×24 pixels ou de 20×15 pixels. Un très grand nombre de caractéristiques par fenêtre est ainsi généré, Viola et Jones donnant l'exemple d'une fenêtre de taille 24×24 qui génère environ 160000 caractéristiques. En phase de détection, l'ensemble de l'image est parcouru en déplaçant la fenêtre de détection d'un certain pas dans le sens horizontal et vertical (ce pas valant 1 pixel dans l'algorithme original [22]). Les changements d'échelles se font en modifiant successivement la taille de la fenêtre de détection. Viola et Jones utilisent un facteur multiplicatif de 1.25, jusqu'à ce que la fenêtre couvre la totalité de l'image.

Finalement, et afin d'être plus robuste aux variations d'illumination, les fenêtres sont normalisées par la variance [22].

La conséquence de ces choix techniques, notamment le recours aux images intégrales, est un gain notable en efficacité, les caractéristiques étant évaluées très rapidement quelle que soit la taille de la fenêtre. (figure 2.4).



$$\text{Somme} = (36+2) - (10+9) = 19$$

Figure 2.4 Exemple de calcul d'une caractéristique Haar

- **Cascade de classifieurs**

La méthode de Viola et Jones est basée sur une approche par recherche exhaustive sur l'ensemble de l'image, qui teste la présence de l'objet dans une fenêtre à toutes les positions et à plusieurs échelles. Cette approche est cependant extrêmement coûteuse en calcul [22].

L'une des idées-clés de la méthode pour réduire ce coût réside dans l'organisation de l'algorithme de détection en une cascade de classifieurs [23].

Appliqués séquentiellement, ces classifieurs prennent une décision d'acceptation, la fenêtre contient l'objet et l'exemple est alors passé au classifieur suivant, ou de rejet, la fenêtre ne contient pas l'objet et dans ce cas l'exemple est définitivement écarté. L'idée est que l'immense majorité des fenêtres testées étant négatives (ne contiennent pas l'objet), il est avantageux de pouvoir les rejeter avec le moins possible de calculs. Ici, les classifieurs les plus simples, donc les plus rapides, sont situés au début de la cascade, et rejettent très rapidement la grande majorité des exemples négatifs [23]. Cette structure en cascade peut également s'interpréter comme un arbre de décision dégénéré, puisque chaque nœud ne comporte qu'une seule branche [23] (figure 2.5).

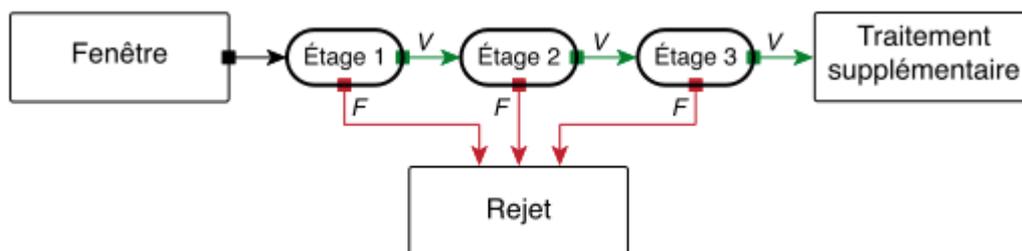


Figure 2.5 Cascade de classifieurs AdaBoost.

En pratique, la cascade est constituée d'une succession d'étages, chacune étant formée d'un classifieur fort appris par AdaBoost. L'apprentissage du classifieur de l'étage est réalisé avec les exemples qui ont passé l'étage n et $n-1$; ce classifieur doit donc faire face à un problème plus difficile : plus on monte dans les étages, plus les classifieurs sont complexes [22].

2.4 Extraction de paramètre

Cette étape elle est importante pour la reconnaissance faciale. L'extraction de paramètre ce fait a partir de la méthode de visage propre (d'eigenface).

2.4.1 Introduction

La reconnaissance faciale est certainement l'un des problèmes les plus populaires de vision par ordinateur ce domaine a été bien étudié au cours des 50 dernières années. Les intentions premiers à explorer la reconnaissance du visage ont été faites dans les années 60 mais il a été jusqu'à les années 90 quand Turk et Pentland mis en œuvre les (visages propres) algorithmes, que ce domaine a donné des résultats très intéressants et utiles [15].

2.4.2 Problématique

La reconnaissance faciale apporte plusieurs problèmes qui sont tout à fait unique à ce domaine et qui en font l'un des plus grands défis dans le groupe des problèmes d'apprentissage automatique.

- **Problème Illumination** : en raison de la réflexivité de la peau humaine, même un léger changement dans l'éclairage de l'image peut largement influencer sur les résultats.
- **Pose des changements** : toute rotation de l'eu d'une personne peut affecter la performance.
- **Temporisation** : bien sûr que, en raison du vieillissement des individus humains, la base de données doivent être régulièrement mis à jour.

2.4.3 Méthodes de la reconnaissance faciale

Il existe plusieurs types de méthodes, qui se distinguent par le type de caractéristiques utilisées [15]:

- Les approches par modèles de visage procèdent à une analyse biométrique des visages pour déterminer des mesures telles que la distance entre les yeux, la longueur du nez, la forme du menton...
- Les approches « image » comparent au contraire directement les visages, en les considérant comme des images, pour lesquelles des mesures de similarité pré-attentives (sans modèle a priori) sont définies.
- Des approches hybrides utilisent les notions de similarité entre images, mais en rajoutant des connaissances a priori sur la structure d'un visage.

2.4.4 Analyse par les visages propres

Le problème que nous allons, ici, considérer est : étant une image de visage donnée on souhaite déterminer l'identité de la personne. Pour ce faire, à chaque visage dit "de référence" sera associé un vecteur de caractéristiques. Ces caractéristiques sont supposées être invariantes pour une même personne et différentes d'une personne à l'autre. Il existe plusieurs types de méthodes pour identifier la personne, qui se distinguent par le type de caractéristiques utilisées. Lors de ce PFE, nous utiliserons une approche, dite "images", qui compare directement les visages en les

considérants comme des images, pour lesquelles des mesures de similarité pré-attentives sont définies [15].

Chaque visage sera donc une image qui pourra être représentée par un vecteur à n composantes et l'ensemble des visages formera un nuage de points dans un espace R_n . On note Ψ la moyenne de ce nuage de points, ou "visage moyen". Il est toujours le même pour une base de références fixée [15].

Le principe de la méthode des Eigenfaces est de modéliser la différence d'un visage quelconque par rapport à ce visage moyen par un ensemble limité d'images u_h , appelées eigenfaces. Une image de visage J est donc exprimée comme le visage moyen auquel s'ajoute une combinaison linéaire d'eigenfaces :

$$J = \Psi + \sum_i w_h u_h + \epsilon \quad \longrightarrow \text{2.1}$$

Ou w_h représente le poids de l'eigenface d'indice h dans le visage J , et ϵ l'erreur entre J et son approximation par les eigenfaces.

2.4.5 Analyse par ACP

C'est l'algorithme de référence de la reconnaissance faciale. C'est une méthode de réduction d'espace connue aussi sous le nom de transformée de Karhunen-Loeveou PCA. Elle a été utilisée par Turk et Pentland dans [15], où la notion de "eigenfaces" a été introduite pour exprimer les différents vecteurs propres de la base d'images de visage de la PCA. En effet, on peut considérer que les images de visage de dimension $N \times N$ pixels ne forment qu'un sous ensemble de l'espace des images de dimension $N \times N$. Il est alors intéressant de chercher une représentation d'un visage dans cet sous ensemble appelé espace des visages. Cet espace est construit à partir d'un ensemble d'image de visage constituant la base d'apprentissage. La PCA a pour objectif de maximiser la variance entre les données [15].

Cela revient à maximiser la variance entre les "différents visages". Pour un ensemble de N images, $\mathbf{X} = (x_1, x_2, \dots, \dots, x_N)$ est l'ensemble des images exemples,

l'analyse par composante principale est une décomposition en éléments propres de la matrice de covariance C de ces données d'apprentissage [15].

$$C = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \longrightarrow (2.2)$$

Où x est la valeur moyenne de X :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \longrightarrow (2.3)$$

Si x_0 est un vecteur de dimension M. On aura alors M vecteurs propres (eigenvectors) (v_1, v_2, \dots, v_M) de dimension M et M valeurs propres $(\lambda_1, \lambda_2, \dots, \lambda_M)$.

Ces vecteurs propres forment une base orthogonale en résolvant l'équation :

$$Cv = \lambda v \longrightarrow (2.4)$$

Où C est la matrice de covariance et v est un vecteur et λ , un scalaire.

Les valeurs propres $(\lambda_1, \lambda_2, \dots, \lambda_M)$ représentent le taux de variation le long de l'axe du vecteur propre associé. Dès lors, on pourra éliminer les vecteurs propres les moins importants sans perte conséquente d'information. On définit alors la proportion P de la variance contenue dans les Q premiers éléments de l'ensemble des vecteurs propres choisis :

$$P = \frac{\sum_{q=1}^Q \lambda_q}{\sum_{i=1}^M \lambda_q} \longrightarrow (2.5)$$

Où $Q < M$.

La projection d'une image I , Ω dans cet espace de visage est donnée par :

$$w_i = (I - \bar{x})^T v_i \longrightarrow (2.6)$$

Où i est la composante selon l'axe v_i

La classification des données est ensuite fournie par un calcul de distance euclidienne ou par des moyens plus élaborés tel que le SVM (Support Vector Machines). Cette approche PCA est assez dépendante de l'espace de visage choisi. En effet, plus la base d'apprentissage est grande et intègre des variations d'illumination, de pose et d'expression mieux on peut présenter les visages dans cet espace. On trouve dans la littérature des essais d'implémentation de cette approche.

Les auteurs ont construit un espace de transformée en ondelette discrète (ondelettes de Daubechies) des visages. Cela permet d'améliorer les performances de la reconnaissance. En effet, une analyse en ondelette du visage permet d'en extraire les informations pertinentes. Ensuite la PCA réduit les redondances de l'analyse. Le classifieur choisi dans ce cas était une distance euclidienne [15] (figure 2.6).



Figure 2.6. Exemple d'eigenfaces générés, comme précédemment, avec 60 images de références [15].

2.4.6 Analyse par module d'apprentissage

Le rôle de ce module est d'organiser les vecteurs de traits en classes et de calculer le seuil de chaque classe. Suivant lequel une décision de reconnaissance sera prise.

A chaque individu correspond une classe. Dans notre cas, elle est composée de deux vecteurs de traits obtenus en projetant les deux images de la même face sur l'espace de traits ; est définie comme:

$$\Omega_{classe}^k = \frac{1}{2} \sum_{i=1}^2 \Omega_i^k \quad k = 1, \dots, N \longrightarrow (2.7)$$

Où N représente le nombre d'individus.

On aura donc dans la base de données un seul *vecteur* représentant chaque individu.

Afin de déterminer si un individu est connu et appartenant à une classe K ou non, on a à déterminer un seuil appelé "Distance de classe des visages", donne comme suit:

$$Det^k = Max_{i=1,2} (\|\Omega_i^k - \Omega_{classe}^k\|^2) \quad k = 1, \dots, N \longrightarrow (2.8)$$

2.5 Base d'apprentissage

Le rôle de ce module est d'organiser les vecteurs de traits en classes et de calculer le seuil de chaque classe. Suivant lequel une décision de reconnaissance sera prise. (figure 2.7).

A chaque individu correspond une classe. Dans notre cas, elle est composée de deux vecteurs de traits obtenus en projetant les deux images de la même face sur l'espace de traits.

On aura donc dans la base donnée un seul *vecteur* représentant chaque individu.

Afin de déterminer si un individu est connu et appartenant à une classe K ou non, on a à déterminer un seuil appelé "Distance de classe des visages", donne comme suit:

$$Det^k = Max_{i=1,2} (\|\Omega_i^k - \Omega_{classe}^k\|^2) \longrightarrow (2.9)$$

Où $k = 1, \dots, N$.



Figure 2.7. Exemple de quelque visage dans la base de données.

2.6 Classification

Dans cette étape une décision doit être prise par le système. C'est celle de déclarer la face test comme connue ou inconnue, c'est une étape de reconnaissance proprement dite.

Les étapes nécessaires à ce stade sont:

- Calcul de l'image différence des les visages de la base de données.
- Extraction du vecteur de traits.
- Reconnaissance de visage par la distance euclidienne :

$$d = VdB[I] - VdE \longrightarrow \text{2.10}$$

$$\text{distEuc} = \text{distEuc} + d^2 \longrightarrow \text{2.11}$$

Où :

d : La différence entre les visages de base de données et visage d'entrée.

distEuc : Distance euclidienne.

2.7 Test et Décision

Après l'obtention de la distance euclidienne peut maintenant faire la décision

Si ($\text{distEuc} < \text{distEuInt}$) alors

$$\text{Seuil} = 1 - \sqrt{(\text{distEuInt} / (nVdB \times n))} / 255 \longrightarrow \text{2.12}$$

Où :

distEuInt : valeur fixe.

Seuil : valeur fixe.

$nVdB$: nombre de visages dans la base de données.

n : nombres des valeurs propres.

Remarque : Nous avons acquis de la valeur de $dist_{EuclInt}$ et le seuil après une série de tests.

Si le seuil obtenu est inférieur à seuil choisi alors le visage d'entrée est connu sino inconnu.

2.8 Conclusion

Nous avons présenté dans ce chapitre, *une* technique de détection et reconnaissance *des* faces basée essentiellement sur les "faces propres", qui demeure encore au stade de la recherche. *Les avantages* que présente cette méthode sont les suivants:

- Très rapide.
- Simple à implémenter.

Cependant, comme toutes les autres méthodes, elle *ne peut être* idéale. Les inconvénients qu'elle présente sont:

- Le taux de reconnaissance décroît lorsque de grands changements apparaissent dans la taille de la face.

L'inclinaison, qui reste à *ce jour un problème fondamental* dans tous les systèmes de reconnaissance de *faces*.

Pour terminer, cette méthode peut donner *de très* bons résultats si les conditions- d'éclairage, d'orientation et d'échelle restent Inchangées.

Dans le chapitre suivant on représenter les testes et les résultats de notre algorithme (eigenfaces).

Chapitre 3 Implémentation, tests et résultats

3.1 Introduction

La dernière étape du processus de développement concerne l'implémentation de l'application en fonction des technologies choisies. Cette étape correspond à la phase de construction du service de synchronisation et de l'application principale. Nous commencerons, tout d'abord, par l'explication de l'algorithme d'eigenfaces, ACP et la bibliothèque d'OpenCV puis la présentation de l'environnement matériel et logiciel utilisés pour développer notre projet de fin d'études.

3.2 Mise en situation

3.2.1 Définition de l'objectif

L'objectif de notre travail consiste à développer une application permettant la reconnaissance des visages des personnes via une caméra webcam. L'entrée de notre application est une séquence vidéo et la sortie est une valeur qui représente le score de comparaison des caractéristiques des visages extraits de la base avec celle qui existe dans la séquence vidéo. Tout d'abord, on décompose la vidéo en un ensemble d'images. Puis, à partir de cet ensemble on cherche le meilleur choix qui correspond à l'image dans laquelle l'algorithme de recherche a détecté les visages ensuite on calcule les distances caractéristiques et on les compare avec celles des images de la base afin

d'obtenir un score de ressemblance. Ce dernier est comparé à un seuil déjà fixé pour vérifier l'existence de la personne recherchée dans la base.

L'application que nous allons développer doit prendre un nouveau visage d'une personne se trouvant devant la webcam et de le comparer avec ceux de la base de données puis repérer cette personne dans la mémoire de l'ordinateur (dans un fichier texte).

3.2.2 Temps de réponse et performances

Le module de détection et reconnaissance faciale s'exécute dans un programme, donc on doit minimiser les calculs afin d'améliorer ses performances.

La conception doit s'orienter depuis le début vers le choix qui permet au système d'être le plus rapide possible (temps réel), et avec un taux de performance élevé.

3.2.3 Environnement matériel

Afin de mener à bien ce projet, il a été mis à notre disposition un ensemble de matériels dont les caractéristiques sont les suivantes : Un ordinateur Acer avec les caractéristiques suivantes :

- Processeur : Intel® CPU T2130 @ 1.86Ghz
- RAM : 1.00 Go de RAM
- Carte graphique : 256 Mb

3.2.4 Environnement logiciel

- **Logiciel de développement :**

Microsoft Builder C++ 6

- **Bibliothèque graphique :**

Open CV (Open Computer Vision)

3.3 La librairie OpenCv

Un des objectifs buts d'OpenCV est d'aider les utilisateurs à construire rapidement des applications sophistiquées de vision à l'aide de simples opérations de vision par ordinateur.

OpenCV (Open Source Computer Vision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, elle a été écrite en C et C++, puis des interfaces ont été développées pour Python, Ruby, Matlab et autres langages.

Initialement écrite en C il y a 10 ans par des chercheurs de la société Intel, OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40 000 membres actifs. C'est la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie.

3.3.1 Présentation d'OpenCV

OpenCV (Open Computer Vision) est une bibliothèque libre de visualisation en temps réel pour le langage C/C++, elle idéalise les optimisations IPP (Integrated Performance Primitives) d'Intel mais fonctionne aussi très bien sans. Les domaines d'utilisation sont variés : robotique, détection et reconnaissance d'objets ou visage ainsi que suivi et étude de leurs mouvements.

3.4 Implémentation

3.4.1 Organisation logicielle

Les programmes décrits ont été développés en C++ Builder 6. Ce langage s'est avéré puissant et offre un graphisme de qualité.

Le tableau suivant montre la différence entre les langages de programmation

C#	JAVA	Perl	MATLAB	C++
Langage interprété	Langage intermédiaire	Langage interprété	Langage interprété	Langage compilé
Utilisé pour les applications avec interfaces graphique.	Utilisé pour les applications avec interfaces graphiques et la programmation orientée internet	L'un des langages les plus utilisés pour la programmation de sites Web interactifs.	Utilisé pour le Calcul mathématique et les solutions nécessitant très grande puissance de calcul.	utilisé pour développer des applications graphiques et pour la programmation système objet.
Disponible sur plusieurs plateformes.	Disponible sur plusieurs plateformes.	Disponible sur plusieurs plateformes.	Disponible sur plusieurs plateformes.	Disponible sur plusieurs plateformes
Parmi les langages les plus rapides	Rapide	Souffrir d'une certaine lourdeur.	Rapide	Très rapide
Nombreuse bibliothèques	Nombreuse bibliothèques	Nombreuses fonctionnalités	Nombreuses fonctionnalités mathématique	Nombreuse Bibliothèques
Plateforme de développement à usage facile. (Visual studio)	Plateforme de développement à usage moyennement facile.	Plateforme de développement à usage facile.	Plateforme de développement à usage facile.	Plateforme de développement à usage facile. (Visual studio)

Surcharge des opérateurs	Pas de surcharge des opérateurs	Surcharge des opérateurs	Surcharge des opérateurs	Surcharge des opérateurs
--------------------------	---------------------------------	--------------------------	--------------------------	--------------------------

Tableau 3.1. La déférence entre les langages des programmations

D'après le tableau comparatif ci-dessus, les deux langages de programmation les plus privilégiés sont JAVA et C++. Cependant, le langage C++ possède plus de souplesse pour la maintenance et le développement que le langage JAVA dans le domaine de l'imagerie et de l'embarqué.

Vu également la rapidité de C++ grâce à la disponibilité d'une large collection d'APIs (Application Programming Interface), ce langage nous semble le plus adéquat pour le développement de cette application et la création de son interface graphique.

3.4.2 Besoins fonctionnels

- Récupérer la vidéo à analyser.
- Découper la séquence vidéo en un ensemble d'images.
- Détecter les visages dans chaque image de la vidéo.
- Calculer la distance euclidienne des images cibles et celles de l'entrée.
- Comparer l'image cible avec les images d'entrée.
- Retourner le résultat de reconnaissance.
- Retourner le taux de ressemblance.

3.4.3 Besoins non fonctionnels

- Fixer le seuil de reconnaissance.
- Configurer les chemins de nos ressources.
- Gérer la base.
- Rapidité du calcul des scores.
- Organisation et indexation de la base.

3.5 Interfaces de l'application

3.5.1 Menu de mot de passe

Cette interface présente la fenêtre de mot de passe une sorte de passerelle pour accéder de passage aux applications du PFE à partir duquel on doit écrire le mot de passe correcte.

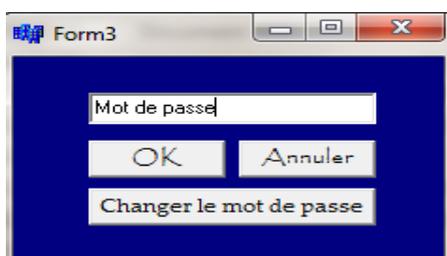


Figure 3.1. Menu de mot de passe

3.5.2 Menu pour modification du mot de passe

La figure 3.2 présente la fenêtre pour modifier l'ancien mot de passe.

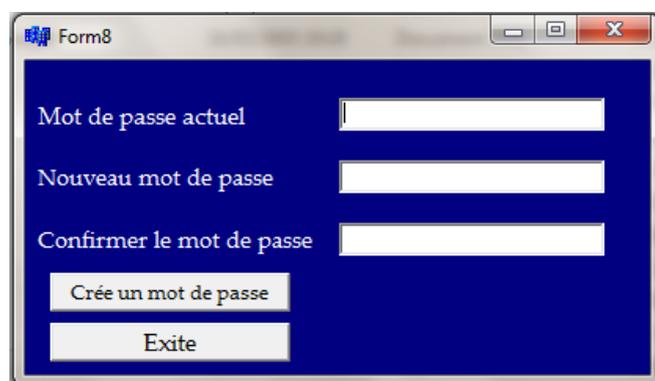


Figure 3.2. Menu modification de mot de passe

3.5.3 Menu principal

La figure 3.3 présente le menu principal une sorte de passerelle pour accéder aux applications du PFE.

Le menu contient des boutons pour la reconnaissance des visages de la caméra, la reconnaissance des visages à partir du fichier, le constructeur de base de données, le propos d'Information et le bouton pour fermer l'application.



Figure 3.3. Menu principal

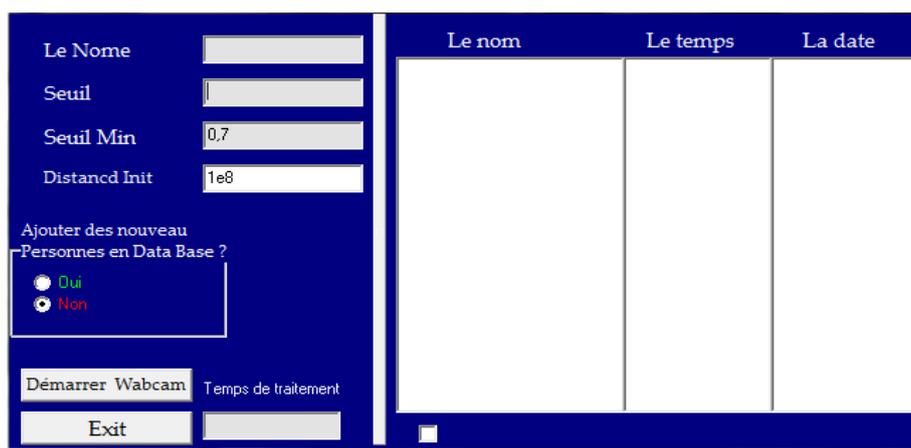
3.5.4 Reconnaissance en temps réel

a la reconnaissance des visages

Les deux options suivantes du menu principal sont dédiées à la reconnaissance des visages. L'option est de reconnaître de l'appareil photo ou d'un fichier. Les deux options ont été réalisées sous la même fonction, la seule différence étant la source des images.

b Reconnaissance via la webcam

Dans L'interface de la reconnaissance en temps réel en peut reconnu les visages en temps réel à partir le bouton (Démarrer Webcam).



The screenshot shows a software interface with a dark blue background. On the left side, there are several input fields: 'Le Nome' (empty), 'Seuil' (empty), 'Seuil Min' (0.7), and 'Distancd Init' (1e8). Below these is a section titled 'Ajouter des nouveau Personnes en Data Base ?' with two radio buttons: 'Oui' (selected) and 'Non'. At the bottom left, there are two buttons: 'Démarrer Wabcam' and 'Exit'. To the right of the input fields is a table with three columns: 'Le nom', 'Le temps', and 'La date'. The table is currently empty. Below the table, there is a 'Temps de traitement' label and an empty input field.

Figure 3.4.Menu de reconnaissance en temps réel.

Tout d'abord, nous récupérons la vidéo à traiter à partir d'une webcam.

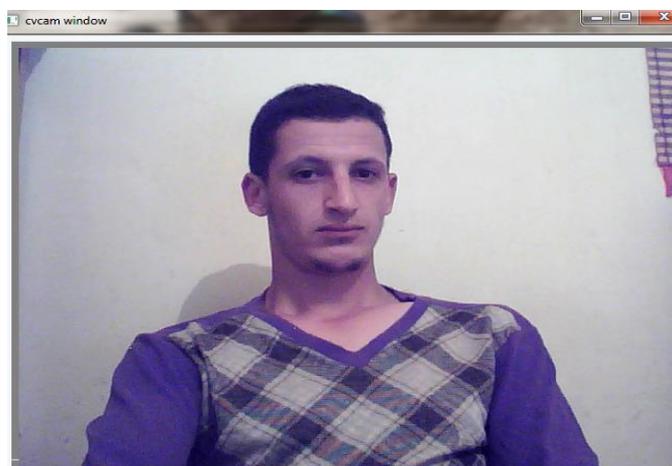


Figure 3.5.Récupération de la vidéo

Puis on peut passer à l'étape de reconnaissance.

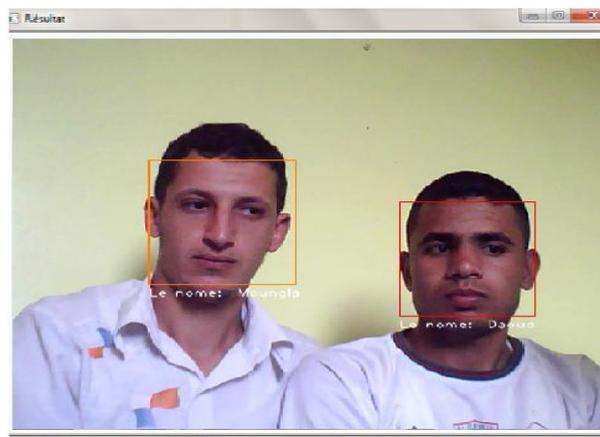


Figure 3.6. Résultat de détection et reconnaissance humain en temps réel.

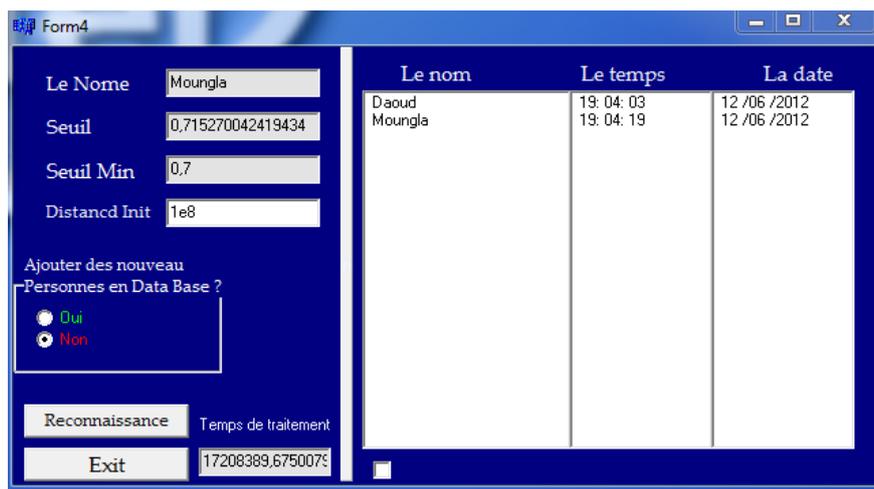


Figure 3.7. Résultat de pointage automatique

3.5.5 Reconnaissance sur une image fixe

L'application devait également une option pour identifier un sujet à partir d'une image ou un fichier vidéo.

Cette partie fonctionne exactement comme l'a été expliqué précédemment. La seule différence était que les images proviennent d'un fichier.

Le menu montre également l'image de la personne reconnue qui est enregistré sur la base de données (voir la figure 3.8).

Le nome	Le temps	La date
---------	----------	---------

Figure 3.8.Détection et reconnaissance humain

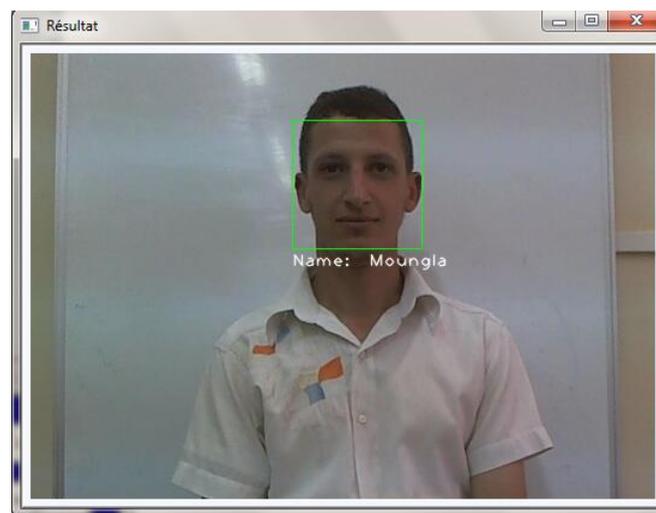


Figure 3.9.Résultat sur image fixe

3.5.6 Base de données

La reconnaissance des visages fonctionne avec une base de données d'images enregistrées précédemment. Cette partie le programme a réussi à créer une base de

données, où les visages des sujets différents ont été identifiés, dans ce cas, en donnant le nom complet.

La demande a pris 10 images de la personne, et a ensuite formé les visages propres pour une utilisation ultérieure dans le processus de reconnaissance.

L'architecture de la base de données intégrée était très simple, il a été divisé en trois différentes colonnes: un numéro d'identité corrélative, le nom de l'objet et le chemin des dix fichiers de chaque face.

Cette partie a également un panneau où il affiche une liste de toutes les personnes sur la base de données on représente ça dans la figure 3.10.

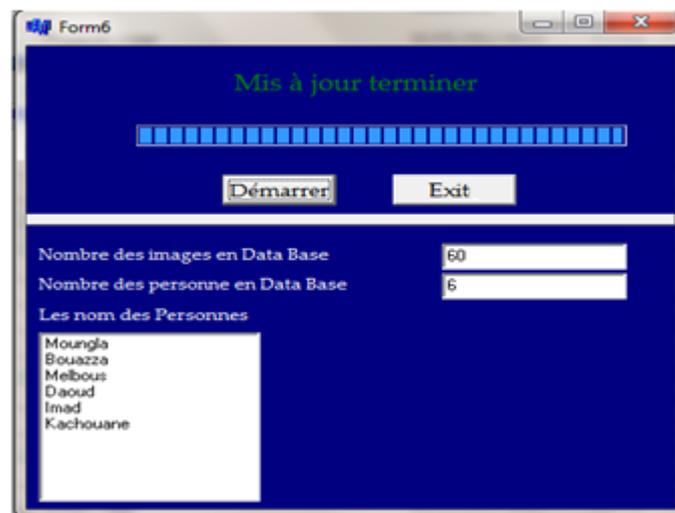


Figure 3.10.Data base

3.5.7 À propos

L'écran À propos d'affiche des informations sur l'application et l'auteur le programme. Cette fenêtre est uniquement informatif, le code n'est utile que dans le joint est le lien e-mail on représente ça dans la figure 3.11.



Figure 3.11. Fiche A propos.

3.6 Tests

L'application a été testée au cours de la recherche ; on 'est basé entièrement sur ces test pour paramétrer nos détecteurs, en essayant d'équilibré pour avoir des taux de détection assez élevé à temps d'exécution minimums.

3.6.1 Situation de test

Les sujets testés ont été assis sur une chaise à environ un mètre de la caméra, qui a toujours été placé sous le même angle.

Tous les tests ont été effectués dans une pièce bien éclairée. L'éclairage a été régulier et blanc, principalement à partir du plafond et de l'avant.

Au moment de l'enregistrement des images dans la base de données, il était important que les projets sous-resté presque immobile, à l'exception de la tête qui a dû être légèrement tourné vers le côté. L'application a découvert la personne et enregistré les images dans la base de données (figure 3.12).

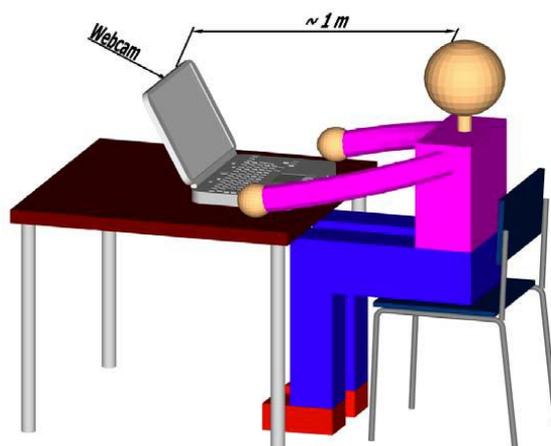


Figure 3.12. Représentation de test situation

3.6.2 Reconnaissance faciale

a Reconnaissance sur image fixe

On n'a testé la méthode d'eigenfaces ; puisqu' on a choisis d'implémenter cette méthode dans notre programme.

Tout ces test on été fait sur une base qu'on a conçu, contenant 100 images de 640 x 480 pixels.

Ces 100 images contiennent 186 visage à détecter et on a pris en compte une détection étant bonne si le cadre de détection couvre plus de la moitié de visage (les figures 3.13, 3.14 et 3.15).



Figure 3.13. Quelques exemples de la base de tests

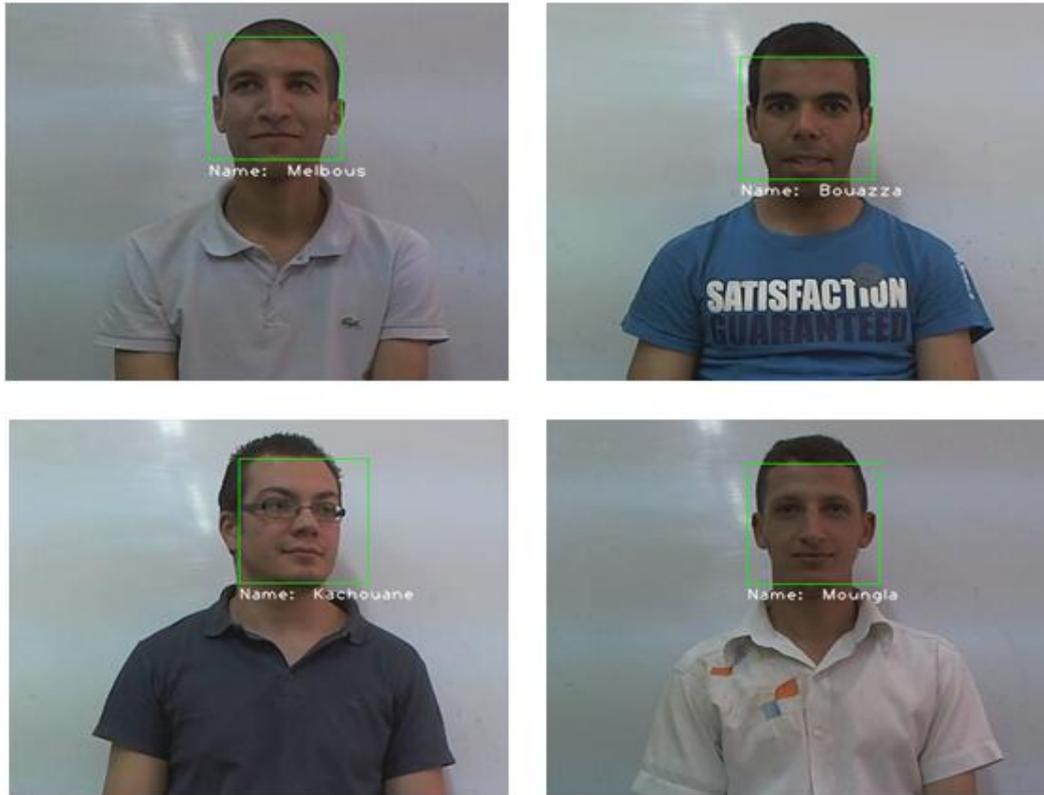


Figure 3.14. Quelques exemples de la reconnaissance avec un seuil = 0.9

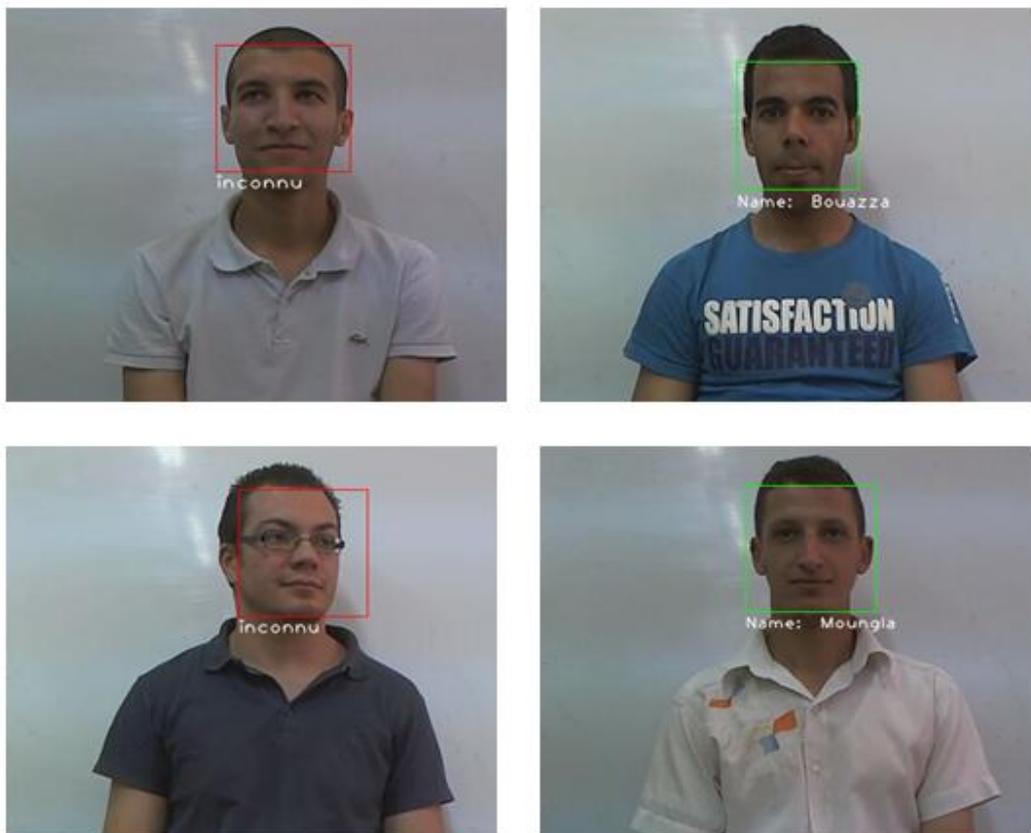


Figure 3.15. Quelques exemples de la reconnaissance avec un seuil = 0.5

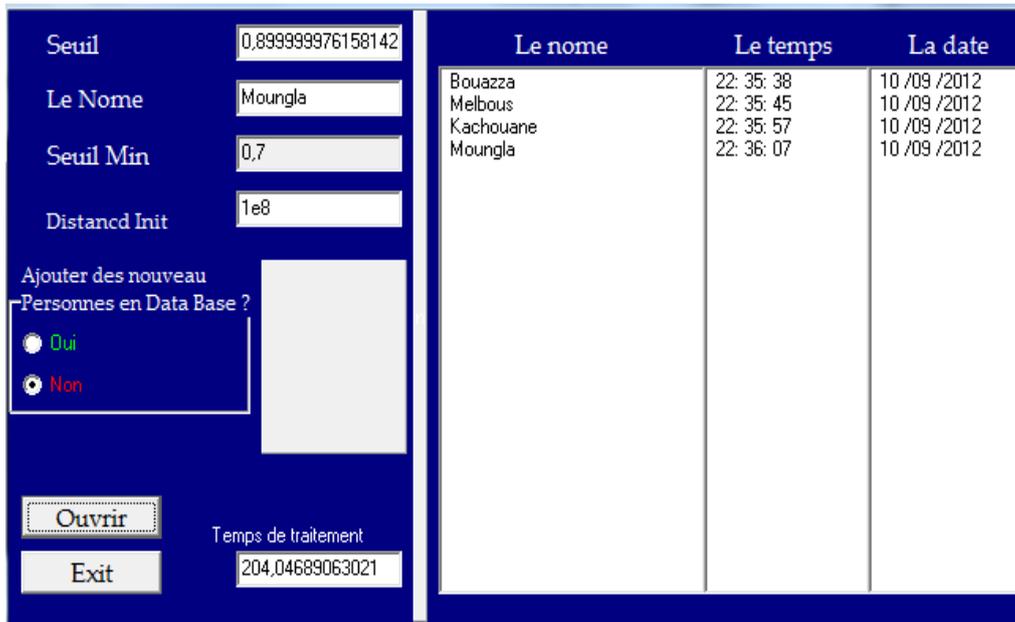


Figure 3.16. Résultat de pointage sur image fixe

b Le seuil utilisé

Le seuil utilisé	Visages connus	Visages inconnus	Temps de la reconnaissance
0	0 (0 %)	186 (100 %)	103
0.1	0 (0 %)	186 (100 %)	105
0.5	98 (53 %)	90 (47 %)	155
0.6	102 (55 %)	84 (45 %)	158
0.7	132 (71 %)	53 (29 %)	179
0.8	171 (92 %)	15 (8 %)	193
0.9	186 (100 %)	0 (0 %)	198
1	186 (100 %)	0 (0 %)	203

Tableau 3.2. Tests de l'algorithme d'eigenfaces avec différent seuils

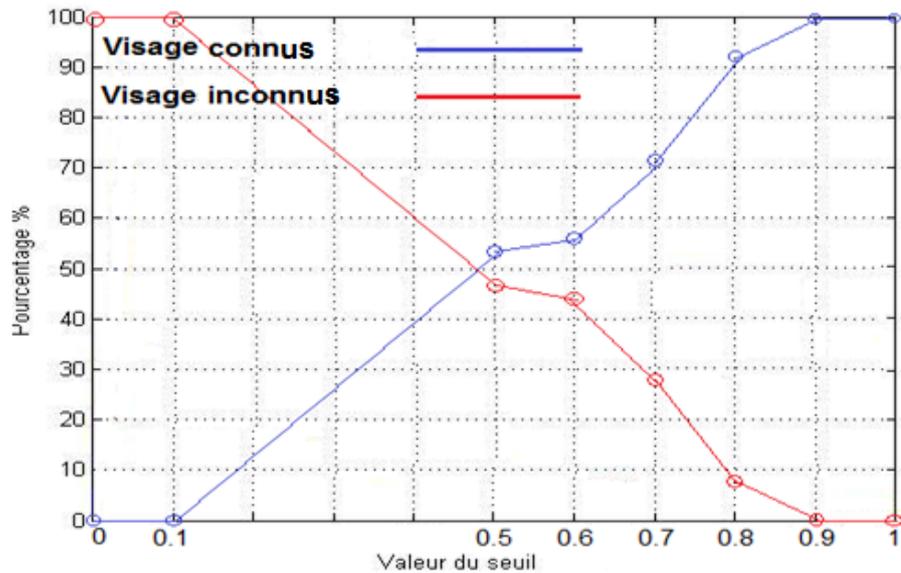


Figure 3.17. Performance de visages propres

On remarque que les meilleurs taux sont le seuil = 0.9 et 1.

c Reconnaissance via webcam

La reconnaissance des visages de l'option caméra est capable de détecter et d'identifier les visages, précédemment enregistrés sur la base de données (voir la figure3.8), à partir d'une source d'imagerie en temps réel. Les visages étaient identifiés et puis le nom de la personne a été visualisé sur l'écran et dans le menu. Puis pointer la personne automatiquement.



Figure 3.18. Résultat de détection et reconnaissance humain en temps réel

Le nom	Le temps	La date
Daoud	19: 04: 03	12 /06 /2012
Moungla	19: 04: 19	12 /06 /2012

Figure 3.19. Résultat de pointage automatique

3.6.3 Résultats des tests

Les résultats des tests étaient favorables; environ 85% des tests ont été satisfaisants. Certains tests ont échoué en raison de la mise en œuvre des problèmes. Ces tests ont été répétés, mais les résultats de l'enquête n'ont pas été touchés (les figures 3.6 et 3.7).

3.7 Conclusion

Le passage de l'algorithme en théorie à l'application réelle était la phase la plus importante. L'apprentissage nous a pris beaucoup de temps, Le système de détection des visages fonctionne avec une précision presque parfaite (99%).L'application est capable de détecter, avec-des problèmes, un certain nombre de visages dans le même temps à partir de n'importe quelle source.

Le système de reconnaissance faciale n'était pas assez solide pour obtenir la reconnaissance de haute accrue. Les principales raisons pour lesquelles ce n'était de la variance du bruit dans les images en raison de l'éclairage et le nombre limité de visages propres qui ont été utilisés pour le PCA transforment. Cependant, les résultats du test étaient favorables, donnant des taux de plus de 90% dans certains

cas. Ainsi, l'approche visages propres satisfait aux exigences de ce projet avec sa vitesse, la simplicité et la capacité d'apprentissage pour la résolution de ce projet.

La base de données a créé le visage sans problème, de sorte qu'il n'était pas nécessaire d'être con-canalisé la poursuite des travaux dans ce domaine.

Les principaux problèmes rencontrés dans la mise en œuvre de l'application ont été l'origine de fuites de mémoire, causée par certaines images qui sont mal déchargés après utilisation. Un autre problème était-le bruit sur les images qui a aggravé la précision de la reconnaissance.

En conclusion, un système robuste face à la reconnaissance doit être insensible à:

- Changements dans l'éclairage
- Changements dans les orientations de la tête et l'échelle
- Présence de détails faciaux tels que les lunettes, la barbe ou d'autres artefacts
- Contexte
- Bruit

Toutefois, l'application créée est tout à fait efficace pour détecter et reconnaître des visages dans des environnements contrôlés.

Conclusion générale

Dû à sa facilité d'utilisation, la reconnaissance des visages, considérée comme étant les fruits de recherches de plusieurs experts dans le domaine de l'imagerie, restera un outil puissant malgré l'existence d'autres méthodes biométriques de reconnaissance. En dépit de l'énorme progression en informatique et plus spécifiquement le traitement d'images, le traitement des données et l'élaboration des bases des données, les systèmes appliquant les techniques de la reconnaissance de visages restent relativement peu convenables. D'ailleurs il reste encore des recherches à mener et des améliorations à effectuer dans ces deux domaines, mais dans les critères choisis pour effectuer le processus de reconnaissance. Pour surmonter les difficultés de la modulation de la morphologie faciale, avec tout ce qu'elle contient comme éléments, il est important que les anthropologues participent plus activement dans le développement de ce genre de systèmes, ce que nous n'avons pas pu constater dans la littérature anthropologique et médico-légale. En effet, la plupart des systèmes qui essaient d'intégrer la morphologie du visage dans leurs algorithmes de reconnaissance sont développés par les informaticiens suivant les méthodes de modulation des objets en général, sans prendre en compte la spécificité du visage humain, ce qui constitue un facteur défavorable pour l'avancement de cette technique. Nous proposons, donc, de fortifier la collaboration entre les spécialistes des deux sciences, bien que très différentes l'une de l'autre dans leur nature et leurs méthodes de recherches, pour aboutir aux meilleurs résultats.

Malgré tous les progrès qui ont été réalisés, les problèmes de pose et d'éclairage, et de l'identification dans des environnements extérieurs restent des challenges qui susciteront les efforts des chercheurs. Actuellement, il y'a une nouvelle tendance qui arrive et qui commence à susciter les efforts c'est le

multimodale, dans lequel on combine plusieurs technologies biométriques, ou plusieurs algorithmes de reconnaissance, et dans lequel on utilise divers classificateurs pondérés pour essayer d'améliorer les performances de reconnaissance et réaliser des campagnes de test complètes pour valider le concept. Parmi les perspectives ouvertes à ce projet, l'utilisation d'autres méthodes de reconnaissance, et la combinaison avec d'autre technologie biométriques comme l'empreinte digitale ou l'iris pour finaliser l'application multimodale et fiabiliser le système en diminuant la sensibilité aux conditions d'éclairage par de nouvelles normalisations.

Installation et Configuration OpenCV 1.1 on Borland C++ Builder 6.0

1 Introduction

OpenCV 1.1 bibliothèque est d'abord développé sous Microsoft Visual Studio plate-forme (fonctionnant sous Microsoft Windows). Par conséquent, tous *. Lib fichiers ont été générés au titre du compilateur Visual Studio.

Borland C + + Builder utilise la méthode différente pour importer des bibliothèques.

Plus d'informations peuvent être renvoyés dans.

Dans ce tutoriel, nous allons expliquer en plusieurs étapes pour installer et configurer OpenCV 1.1 sur Borland C + + Builder.

Ce tutoriel est principalement basé et inspiré par [20] et [21]. Nous allons également expliquer quelques trucs pour surmonter le conflit entre highgui.dll et plusieurs bibliothèques ActiveX, lorsque les deux d'entre eux sont utilisés simultanément pour effectuer un traitement vidéo en temps réel.

2 Installation

Tout d'abord, nous avons téléchargé OpenCV 1.1 à partir de Source forge, comme le montre la Fig.1 ci-dessous. Installer OpenCV 1,1 en répertoire:

C: Program Files \ \ OpenCV1.1

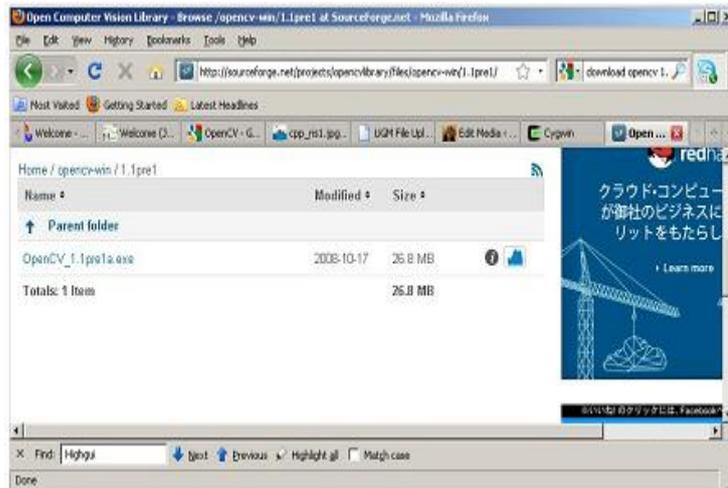


Figure.1. Télécharger OpenCV_1.1pre1a.exe de Source forge

3. Configuration

Inclure tous les 1.1 OpenCV *. Lib fichiers de votre projet BCB. Pour ce faire, cliquez sur Afficher les gestionnaire de projet et droit cliquez sur votre fichier de projet, comme indiqué à la Figure 2 ci-dessous.

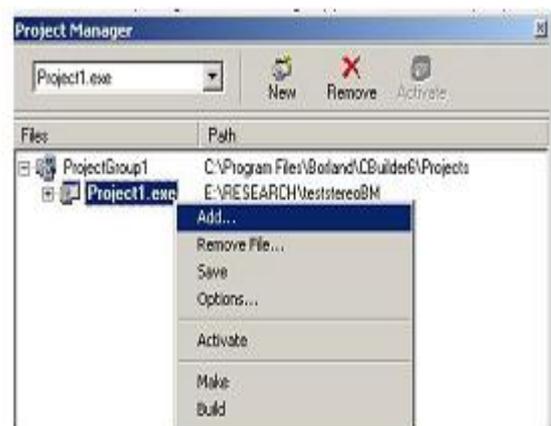


Figure.2. Ajout à la bibliothèque de BCB projet

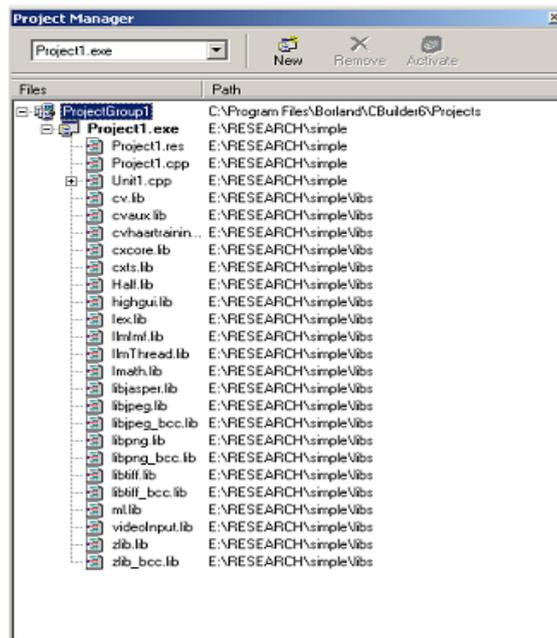


Figure.3. Include dans la bibliothèque BCB projet

Toutes les bibliothèques incluses sont présentés dans la figure 3 ci-dessus. Ensuite, nous devrions définir le chemin vers le fichier d'en-tête de plusieurs en relation avec ces bibliothèques incluses sont présentés dans la figure 4. Pour ce faire, ouvrez "Inclure la trajectoire de" fenêtre en cliquant sur:

Project → Options → Directories/Conditionals → Include Path.

C:\Program Files\OpenCV1.1\cv\include

C:\Program Files\OpenCV1.1\cvaux\include

C:\Program Files\OpenCV1.1\cxcore\include

C:\Program Files\OpenCV1.1\ml\include

C:\Program Files\OpenCV1.1\otherlibs_graphics\include

C:\Program Files\OpenCV1.1\otherlibs\ffopencv

C:\Program Files\OpenCV1.1\otherlibs\highgui

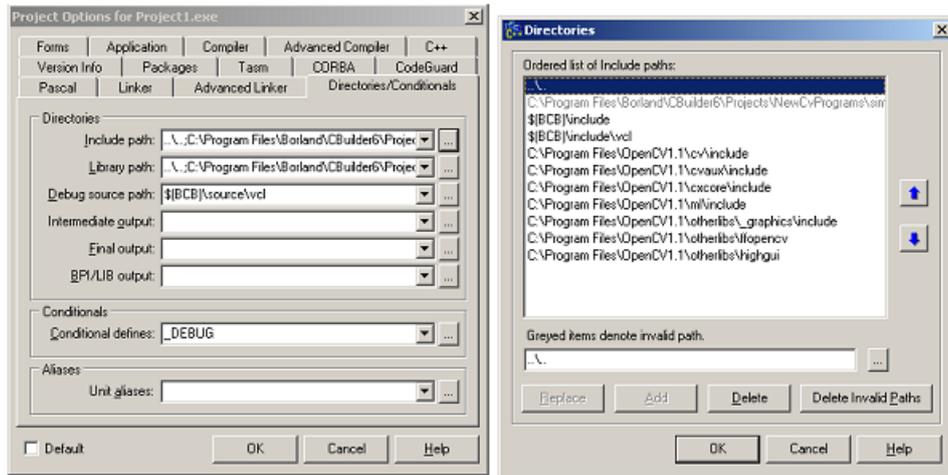


Figure.4. Path configuration in BCB project

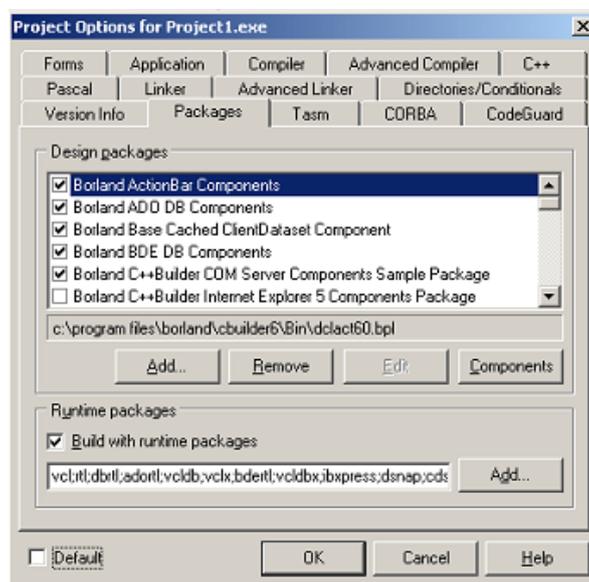


Figure.5. Construire avec les packages d'exécution

4. En utilisant OpenCV 1.1 avec ActiveX Library

Lorsque vous utilisez la bibliothèque ActiveX pour capturer et traiter l'image de votre appareil photo, vous devez configurer votre dll OpenCV puisque les bibliothèques de charge highgui.dll OLE avant votre ActiveX fait. Ce résultant normalement un message d'erreur dans votre application.

OLE est une technologie qui permet à une application pour créer des documents qui contiennent des composés des informations d'un certain nombre de sources différentes. Par exemple, un document dans un mot OLE-activé processeur peut accepter un objet feuille de calcul incorporée. Contrairement aux traditionnels "copier-coller" des méthodes lorsque la demande de réception de modifier le format des informations collées, les documents incorporés conservent toutes leurs propriétés originales. Si l'utilisateur décide de modifier les données intégrées, Windows active l'application d'origine et les charges du document incorporé.

Pour configurer votre application, cliquez sur

Project → Options → Directories/Conditionals Include → Path

Et ouvert "Linker Avancé", comme indiqué dans la Figure 7 et procéder à "Retard charge" section. Ajouter cv110.dll, cxcore110.dll, et highgui110.dll comme «DLL pour retard-charge". Maintenant, essayez de recompiler votre application.

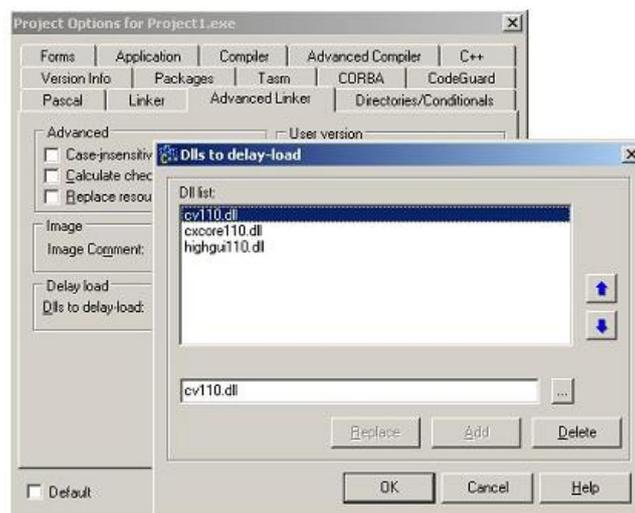


Figure.6. Retarder certains fichiers dll

Bibliographie

- [1] R.C. GONZALES et P. WINTZ : 'Digital Image Processing ', Addison Wesley, 1997
- [2] M. ANDER : ' Introduction aux techniques de traitement d'images ', Eyrolles, 1987.
- [3] MOISE MWEZE : ' Compression Des Images', TFC/ISP Bukavu, 2003-2004
- [4] BOUABIDA Naima : ' Segmentation des tissus cérébraux sur des images par résonance Magnétique ', Projet de fin d'études, Electronique, ENP, Alger, 2007.
- [5] FRADJ Mohammed Karim, LOUKAL Abdelkader : ' Segmentation par étiquetage des artères coronaires avec implémentation sur DSP C6000 ', Projet de fin d'études, Electronique, ENP, Alger, 2007.
- [6] MICROSOFT, ' Encyclopédie Encarta ', 2005.
- [7] A. d'HARDANCOURT : ' Fou du multimédia 'Sybex 1995
- [8] R. BOUABBANA. M. BOURAOUI : ' Conception et réalisation d'un OCR multitalle ', Mémoire de master, électronique, USDB, BLIDA 2009.
- [9] S. PHILIPP J. COCQUEREZ : ' Analyse d'image : filtrage et segmentation ' Éditions MASSON, 1995.
- [10] Hadjila Feth Allah et Bouabdallah Réda : ' Reconnaissance des visages en utilisant les réseaux de neurones', Mémoire d'ingénieur. Université de Tlemcen. 2003.
- [11] Azriel Rosenfeld : 'Digital straight line segments', IEEE Transaction on computers,

Vol.C-23, n°12, December 1974.

[12] S.A. Sirohey, C.L. Wilson, and R. Chellappa : ' Human and machine recognition of faces' , A survey. 1994.

[13] P. Viola, M. Jones : ' Rapid Object Detection using a Boosted Cascade of Simple Features' , Conference on Computer Vision and Pattern Recognition 2001.

[14] Matthew Turk and Alex Pentland : ' Eigenfaces for recognition' , J. Cognitive Neuroscience, 3(1) :71_86, 1991.

[15] M.A. Turk and A.P. Pentland : ' Face recognition using eigenfaces. Computer Vision and Pattern Recognition' , 1991. Proceedings CVPR '91., IEEE Computer Society Conference on, pages 586591, Jun 1991.

[16] by Gary Bradski and Adrian Kaehler : ' Learning OpenCV' , September 2008 : First Edition.

[17] RobertLaganière : ' OpenCV 2 Computer Vision Application Programming Cookbook' , First published: May 2011.

[18] OpenCV with C++ Builder. Available: <http://opencv.willowgarage.com/wiki/C%2B%2BBuilder>.

[19] OpenCV 1.1, creating new project in Borland C++ Builder. Available : http://www.compvision.ru/wiki/%D0%94%D0%BB%D1%8F_OpenCv_1.1pre1 (Russian Language) .

[20] OpenCV Installation. Available: <http://www.compvision.ru/forum/index.php?showtopic=2> (Russian Language) .

[21] OLE Concepts and Requirements Overview. Available: <http://support.microsoft.com/kb/86008>.

[22] Yannick Benezeth : ' Détection de la présence humaine par vision' , Thèse Université d'Orléans, page 31, Octobre 2009.

[23] Paul Viola et Michael Jones : ' Robust Real-time Object Detection', IJCV,workshop on SCTV, Vancouver, Canada, Juillet 2001.