



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Modèle Intelligent et Décision(M.I.D)

Thème

Détection et protection des visages en utilisant la bibliothèque OpenCV

Réalisé par :

- *M^{elle} Khater khadidja*

Présenté le 24 Juin 2015 devant le jury composé de MM.

- *M^{me} LARBAOUI.N* (Président)
- *M^r .BENAISSA.M* (Encadreur)
- *M^r BENMOUNA.Y* (Examineur)
- *M^r BELHOUCINE.A* (Examineur)

Année universitaire : 2014-2015

Remerciements

Louange à **الله**, seigneur de l'univers.

Je remercie toutes les personnes ayant contribué et faciliter la réalisation de ce travail dans de bonnes conditions.

Je remercie plus particulièrement.

Ma famille et mes amis pour l'amour et le soutien qu'ils m'ont toujours accordés

Monsieur Benaissa Mohamed qui il m'a guidé avec grande résignation tout au long de la préparation de ce travail. Merci pour tous les conseils, et les bienveillances.

Je remercie vivement les membres de jury, d'avoir accepté de juger mon travail.

Je remercie à l'ensemble des personnels du département de l'informatique à l'université Abou Berk Belkaid.

Merci à tous et à toutes

Dédicaces

A

*- La mémoire de mon grand-père
qui est un père pour moi*

- Tous ceux que j'aime

- Tous ceux qui m'ont aidé

Khadidja

الكشف عن الوجه و مكوناته (العينين و الفم والأنف الخ) زادت اهتماما في الآونة الأخيرة بالنظر إلى عدد وافر من التطبيقات الناتجة في التواصل بين الإنسان و الآلة.

الهدف من هذا المشروع هو الكشف عن الوجوه و العيون في صورة رقمية أو مشهد فيديو تم التقاطه بواسطة كاميرا ويب باستخدام OpenCV المكتبة المرئية التي توفر أدوات البرمجة الحرة لمعالجة البيانات في شكل صورة أو ملفات فيديو، ويتم تنفيذها في خوارزمية فيولا و جونز التي تعتبر سهلة الاستخدام في اخفاء العيون المكشوف عنها في صورة رقمية، و طمس الوجوه المكشوف عنها في الفيديو.

الكلمات المفتاحية: الكشف عن الوجه، طمس الوجوه، اخفاء العيون، OpenCV، خوارزمية فيولا و جونز

Résumé

La détection de visages et de composantes faciales (les yeux, la bouche, le nez.....etc) a suscité récemment un intérêt grandissant vu la multitude d'applications qui en découlent en communication homme machine.

L'objectif de ce projet est la détection des visages et des yeux dans une image numérique ou dans une scène vidéo capturée par une webcam en utilisant la bibliothèque visuelle OpenCV qui offre gratuitement des outils de programmation pour traiter des données en format image ou des fichiers vidéo. Les visages et les yeux détectés par la méthode de Viola et Jones seront automatiquement protégés et brouillés.

Mots clés : détection du visage, brouillage du visage, l'escompte des yeux, OpenCV, l'algorithme de Viola Et Jones.

Abstract

Face detection and facial components (eyes, mouth, nose etc) recently increased interest given the multitude of applications resulting in Human Computer Interface.

The objective of this project is the detection of faces and eyes in a digital image or a video's scene captured by a webcam using visual library OpenCV witch offers free programming tools to process data in image format or files video. The faces and the detected by Viola and Jones method will be automatically protected and scrambled.

Keywords: face detection, blurring of faces, cover the eyes, OpenCV, the Viola and Jones's algorithm.

Table des matières

Introduction générales.....	10
------------------------------------	-----------

Chapitre I : Présentation d'OpenCV

1. Introduction	12
2. Présentation d'OpenCV	12
2.1. Fonctionnalités	12
2.1.1. Traitement d'images	12
2.1.2. Traitement Vidéos.....	12
3. Structure de la librairie OpenCV	13
4. Installation et configuration avec Visual Studio 2008.....	14
4.1. Mise en place variables d'environnement.....	14
4.2. Microsoft Visual Studio 2008	14
4.3. Configuration de Visual Studio 2008.....	15
4.3.1. Configuration générale	15
4.3.2. La configuration nécessaire	16
5. Conclusion.....	17

Chapitre II : Généralités sur les images et les Vidéos

1. Introduction	18
2. Définition d'image.....	18
3. Image numérique	18
4. Caractéristiques d'une image numérique	18
4.1. Pixel «Picture Element»	19
4.2. La définition et la résolution d'une image	19

4.3.	Histogramme	20
4.4.	Contours et textures.....	20
4.5.	Luminance.....	21
4.6.	Contraste	21
4.7.	Profondeur.....	21
4.8.	Le poids de l'image.....	22
4.9.	Bruit.....	22
5.	Représentation des couleurs	22
5.1.	L'image monochrome (binaire)	22
5.2.	L'image en niveaux de gris	23
5.3.	L'image en couleurs.....	23
6.	Les formats d'image.....	24
7.	Compression des images	26
7.1.	L'objectif de la compression des images	26
7.2.	Caractérisation de la compression.....	26
7.3.	Les types des méthodes de compressions des images.....	26
7.3.1.	Compression physique et compression logique.....	26
7.3.2.	Compression symétrique et asymétrique	26
7.3.3.	Compression adaptif, semi adaptif et non adaptif.....	27
7.3.4.	Compression sans pertes et la compression avec pertes :.....	27
8.	Définition d'une vidéo numérique.....	29
9.	Les formats vidéo	29
9.1.	Fichier Conteneur.....	29
9.2.	Le piège du pixel flou	29
10.	Les formats de fichiers vidéo.....	31
11.	Représentation d'une séquence vidéo	32
11.1.	La résolution en luminance	33

11.2.	La résolution spatiale.....	33
11.3.	La résolution temporelle.....	33
12.	Compression de la vidéo numérique.....	33
12.1.	Codec.....	34
12.2.	Algorithmes de compression Vidéo	34
12.2.1.	La compression d'images (Technique intra-image)	34
12.2.2.	La compression vidéo (Technique inter-image).....	36
13.	Conclusion	38

Chapitre III : Détection des visages

1.	Introduction	39
2.	Définition de détection des visages	39
2.1.	Application.....	40
3.	Les caractéristiques du visage	41
3.1.	Détection des yeux	41
3.2.	Détection des bouches.....	41
3.3.	Détection de nez.....	42
4.	Les méthodes de détection des visages	42
4.1.	Méthode basée sur les caractéristiques invariables (Feature Invariant).....	42
4.1.1.	Caractéristiques du visage (Facial Feature)	43
4.1.2.	Texture.....	44
4.1.3.	Couleur de la peau (Skin Color)	44
4.2.	Approche basée sur la reconnaissance	46
4.3.	Appariement de gabarits (Template Matching)	46
4.3.1	Des faces prédéfinies de visages	46
4.3.2	Modèle déformable.....	47
4.4.	Méthode basée sur les apparences (Appearance-based method)	47
5.	Algorithme de Viola et Jones	47

5.1.	Caractéristiques pseudo-haar	48
5.2.	Image Intégrale.....	49
5.3.	Sélection de caractéristiques par boosting	50
5.4.	Cascade de classifieurs.....	51
5.5.	Etapes clés.....	52
5.5.1.	Apprentissage.....	52
5.5.2.	Détection.....	53
5.6.	Performance	53
5.7.	Applications	54
6.	Conclusion.....	54

Chapitre IV : Implémentation et Résultats

1.	Introduction	55
2.	L'objectif de travail	55
3.	Environnement du travail	55
3.1.	Environnement matériel ou le hardware	56
3.2.	Environnement immatériel ou le software	56
4.	Implémentation.....	56
4.1.	Les principes de la détection du visage	56
4.1.1.	Cascade (Classifieur)	56
4.1.2.	Les opérations pour lire les images ou les vidéos.....	57
4.1.3.	La fonction de traitement et de détection.....	58
4.1.4.	Le traçage de rectangle autour du visage	59
4.1.5.	La fenêtre du résultat	59
4.2.	Test et résultat de la détection du visage.....	60
4.3.	Les principes de la détection des yeux	61
4.3.1.	Cascade (Classifieur)	61
4.3.2.	La fonction de détection des yeux	61

4.3.3. Le traçage de rectangle autour de l'œil.....	61
4.4. Test et résultat de la détection des yeux.....	62
4.5. Cacher les yeux	63
4.6. Brouillage du visage.....	64
5. Conclusion.....	65
Conclusion générale.....	66
Références.....	67

Listes des figures

Figure I. 1: Configuration général de Microsoft Visual Studio 2008 pour OpenCV	15
Figure I. 2: Configuration de Microsoft Visual Studio 2008 pour la librairie OpenCV	16
Figure I. 3 : Configuration nécessaire de Microsoft Visual Studio 2008 pour OpenCV.....	17
Figure II. 1: L'image comme un groupe de pixels.	19
Figure II. 2: Histogramme obtenu pour l'image	20
Figure II. 3: Une image binaire	22
Figure II. 4: Image en niveaux de gris	23
Figure II. 5: Image en couleurs.....	23
Figure II. 6: compression de type symétrique.....	27
Figure II. 7: Modèle de pixel informatique est carré	30
Figure II. 8 : Modèle de L'affichage sans déformation	30
Figure II. 9: Exemple de technique intra-image par transformation des couleurs.	35
Figure II. 10 : Exemple de technique intra-image par Elimination les redondances.	36
Figure II. 11: Exemple sur la technique Inter-image par le codage différentiel	36
Figure II. 12: Exemple sur la technique Inter-image par La compensation de mouvement	37
Figure II. 13: Séquence type avec des images I, B et P	38
Figure III. 1: Exemple d'images comportant un seul visage	39
Figure III. 2: Exemple d'images comportant plusieurs visages	40
Figure III. 3: Système de couleur HSV.....	45
Figure III. 4: Image convertie à l'espace de couleurs HSV	45
Figure III. 5: Exemple de caractéristiques pseudo-Haar	48
Figure III. 6: La valeur de l'image intégrale au point (x, y)	49
Figure III. 7: Illustration de l'architecture de la cascade : les fenêtres sont traitées séquentiellement par les classifieurs, et rejetées immédiatement si la réponse est négative (F)	52

Figure III. 8: Courbe du taux de détection correct en fonction du nombre de faux positifs	53
Figure IV. 1 : détection du visage dans une image.....	60
Figure IV. 2 : détection du visage dans une vidéo.....	61
Figure IV. 3 : détection des yeux dans une image.....	62
Figure IV. 4: détection des yeux dans une Vidéo.....	63
Figure IV. 5: Cacher les yeux.....	64
Figure IV. 6: Brouillage du visage.....	65

Listes des tableaux

Tableau II.1 : Les différents formats d'image.....	25
--	----

Liste des sigles et des abréviations

3D	Images A T rois D imensions.
API	A pplication P rogramming I nterface
AVI	A udio V ideo I nterleave.
BMP	B it m ap
BSD	B erkeley S oftware D istribution
CD	C ompact D isc
Codec	E ncodeur/ D éc o deur
GIF	G raphical I nterchange F ormat
DPI	D ots P er I nch
Dv	D igital V ideo
DVD	D igital V ertical D isk
Flv	F lash V ideo
FPS	F rames P er S econd
HD	<i>High-Definition</i>
HDTV	<i>High-DefinitionTeleVision</i>
HSV	H ue, S aturation, V alue
IDE	I ntegrated D evelopment E nvironment
IHM	I nterface H omme- M achine
JPEG	J oint P hotographic E xperts G roup
MacOS	M acintosh O perating S ystem
Mov	Quicktime M ovie
MP4	M PEG- 4 P art 14
MPEG	M otion P icture E xpert G roup
NTSC	N ational T ele V ision S ystem C ommette
OPENCV	O pen S ource C omputer V ision
PAL	P hase A lternation L ine
Pixel	P icture E lement
PNG	P ortable N etwork G raphics
PPP	P oints P ar P ouce
PSD	P hoto S hop D ocument
RGB	R eed G reen B lue
RLE	R un L ength E ncoding
RVB	R ouge V ert B leu
SECAM	S équentiel C ouleur A Mémoire

SURF	S peeded U p R obust F eatures
TIFF	T agged I mage F ile F ormat
YCbCr	Luminance (<i>Y</i>), Chrominance (R ed- Y ellow); C hrominance (B lue- Y ellow)
YIQ	(Y :Luminance),(I :In-phase),(Q :Quadrature)
VCD	V ideo C D
VHS	V ideo H ome S ystem
VLC	V ideo L an C lient
VOB	V ideo O bject File
WMV	W indows M edia V ideo
XML	e X tensible M arkup L anguage

INTRODUCTION GENERAL

Introduction générales

Nous vivons actuellement dans l'ère de la technologie et nous essayons d'attribuer les facultés et les capacités humaines aux machines. L'avance scientifique est telle que l'intelligence artificielle est utilisée pour gérer d'une manière optimale des systèmes et des équipements complexes afin de les aider à prendre des décisions pertinentes. Pour parvenir à un tel résultat on doit passer par la détection des formes, des objets et des êtres humains dans leurs milieux naturels et réels.

L'image du visage est un avantage externe incontestable pour l'identification des individus autour de nous. L'être humain a des capacités naturelles pour reconnaître et différencier les visages, car notre cerveau très évolué peut mémoriser un nombre important d'individus avec les différentes caractéristiques. En est-il de même pour une machine ? Ceci définit la problématique de la reconnaissance automatique de visage, qui a engendré un grand nombre de travaux de recherche au cours des dernières années.

La détection de visage est une nouvelle technologie informatique qui détermine les emplacements et les dimensions des visages humains dans des images numériques ou dans une vidéo. Elle détecte les dispositifs faciaux et ignore toute autre chose, tel que des bâtiments, des arbres, des corps et tout dispositif autre que le visage ...etc. Cette technologie est utilisée dans plusieurs domaines comme la biométrie pour l'identification et la reconnaissance faciale, elle est également employée dans les systèmes de surveillance visuelle ainsi que plusieurs appareils photo numériques récentes qui emploient la détection du visage pour la mise au point automatique.

La question qui se pose en ce moment est : Comment parvenir à détecter le visage d'une personne ensuite détecter les yeux afin de les cacher et cela dans une image numérique ou une vidéo?

Le but de ce travail de fin d'étude est :

La détection des visages ainsi que les yeux dans une image numérique ou dans une scène vidéo capturée par une webcam en utilisant la bibliothèque visuelle open-cv (utilisation Visual studio). Les yeux détectés seront automatiquement cachés en utilisant la technique de brouillage (scrambled).

Introduction Générale

Notre mémoire est structuré comme suite :

- Dans le premier chapitre, nous parlons de généralités sur la bibliothèque OpenCV. Dans le présent chapitre, nous décrivons cette bibliothèque en donnant son Présentation, sa structure et sa configuration dans le système d'exploitation Microsoft Windows avec Visual Studio 2008.
- Le deuxième chapitre comprend des généralités sur les images numériques et les vidéos.
- Dans le troisième chapitre nous présentons les méthodes de détection des visages ainsi que le fameux algorithme de Viola et Jones fonctionnant en temps réel.
- En ce qui concerne le quatrième chapitre, il représente la mise en place de deux applications en utilisant la bibliothèque OpenCV.

La première, fait la détection du visage et des yeux dans une image numérique afin de cacher les yeux.

La deuxième, fait la détection du visage et les yeux dans une vidéo capturée par la webcam de l'ordinateur ensuite brouiller le visage.

Chapitre I

Présentation d'OpenCV

1. Introduction

Dans cette partie, nous allons découvrir les notions de base de la bibliothèque OPENCV et sa configuration sous Windows avec Microsoft Visual Studio 2008. OPENCV sera utilisée dans notre projet pour la détection du visage dans une image ou dans une vidéo capturée par une webcam.

2. Présentation d'OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plates-formes Windows, GNU/Linux, Android et MacOS.

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 40000 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 [28].

2.1. Fonctionnalités

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

2.1.1. Traitement d'images

Elle propose la plupart des opérations classiques en traitement bas niveau des images :

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs [29].

2.1.2. Traitement Vidéos

Elle met également à disposition de l'utilisateur quelques fonctions d'interfaces graphiques, comme les curseurs à glissière, les contrôles associés aux événements souris, ou bien l'incrustation de texte dans une image.

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que:

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra).
- Détection de droites, de segment et de cercles par transformée de Hough.
- Détection de visages par la méthode de viola et JONES.
- Détection de mouvement, historique du mouvement.
- Détection de points d'intérêts [29].

3. Structure de la librairie OpenCV

- **Core**: les fonctionnalités de base.

Cette bibliothèque permet de manipuler les structures de base, réaliser des opérations sur des matrices, dessiner sur des images, sauvegarder et charger des données dans des fichiers XML...

- **Imgproc**: traitement d'image.

Nous entrons dans le cœur du sujet. Les fonctions et structures de ce module ont trait aux transformations d'images, au filtrage, à la détection de contours, de points d'intérêt...

- **Features2d**: descripteurs.

Ce module concerne principalement l'extraction de descripteurs selon deux approches courantes (SURF et Star Detector).

- **Objdetect**: détection d'objets.

Cette bibliothèque permet de faire de la reconnaissance d'objets dans une image au moyen de l'algorithme Adaboost (Viola & Jones, 2001) [20].

- **Video** : traitement de flux vidéo.

Ces fonctions servent à segmenter et suivre les objets en mouvement dans une vidéo.

- **Highgui**: entrées-sorties et interface utilisateur.

OpenCV intègre sa propre bibliothèque haut-niveau pour ouvrir, enregistrer et afficher des images et des flux vidéo. Celle-ci contient aussi un certain nombre de fonctions permettant de réaliser des interfaces graphiques très simples, mais largement suffisantes pour tester nos programmes.

- **Calib3d** : calibration, estimation de pose et stéréovision.

Ce module contient des fonctions permettant de reconstruire une scène en 3D à partir d'images acquises avec plusieurs caméras simultanément [1].

4. Installation et configuration avec Visual Studio 2008

4.1. Mise en place variables d'environnement

- Vous devez ajouter le répertoire « C:\opencv\build\x64\vc09\bin » à votre PATH système. Ce répertoire contient les DLL OpenCV nécessaires pour exécuter votre code.
- Si votre type de système est 32 bits remplacer x64 par x86.
- (C:\opencv\build\x86\VC09\bin). A noter la partie de VC09 du chemin est la version de Visual Studio installé sur le système (VC09 = Microsoft Visual Studio 2009).

4.2. Microsoft Visual Studio 2008

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçu par Microsoft.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++ et Visual C# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages [30].

4.3. Configuration de Visual Studio 2008

Pour réaliser la configuration de Visual studio il faut procéder à deux étapes :

- La configuration générale, qui est à faire une seule fois.
- La configuration nécessaire pour chaque nouveau projet.

4.3.1. Configuration générale

Il s'agit d'indiquer à Visual Studio les nouveaux répertoires où sont disponibles les fichiers .h et .lib d'OpenCV.

- Choisir All Configuration pour Configuration.
- Choisir (x64) pour platform (si le type de système d'exploitation 64 bits) ou bien win32 (si le type de système d'exploitation 32bits).
- Dans le volet de gauche, cliquez sur Cofiguration-----> propriétés-----> VC++ Directories----->general
- Dans Additional include Directories ajouter les sous répertoires suivants :
 - ❖ C:\opencv\build\include
 - ❖ C:\opencv\build\include\opencv
 - ❖ C:\opencv\build\include\opencv2

Comme sur la figure I.1

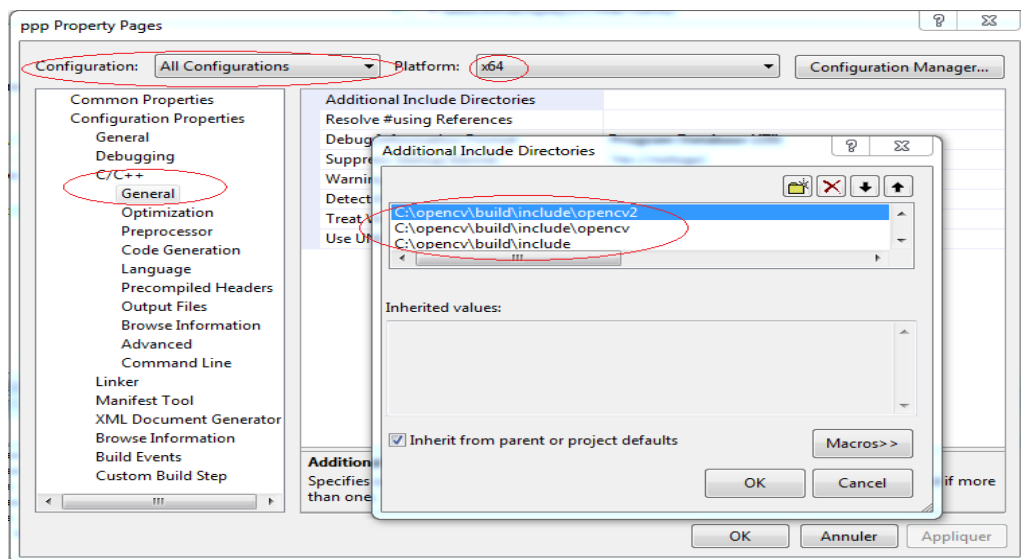


Figure I. 1: Configuration général de Microsoft Visual Studio 2008 pour OpenCV

- Choisir ensuite configuration propriétés ---> Linker ---> General
- Dans Additional Library Directories ajouter la ligne
<<C:\OpenCV\build\x64\VC09\lib>>Comme sur la Figure I.2

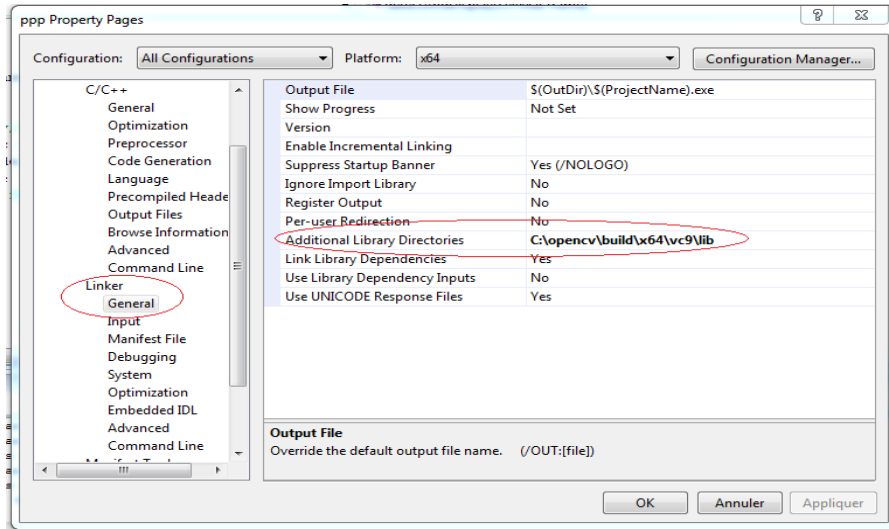


Figure I. 2: Configuration de Microsoft Visual Studio 2008 pour la librairie OpenCV

- **Remarque :** le chemin C:\OpenCV peut varier en fonction de ce que vous avez choisi lors de l'installation.

4.3.2. La configuration nécessaire

Cette étape est à recommencer pour chaque nouveau projet.

- Aller dans Additional Dependencies ajouter les fichiers lib suivants :
 - ❖ opencv_calib3d231d.lib
 - ❖ opencv_contrib231d.lib
 - ❖ opencv_core231d.lib
 - ❖ opencv_features2d231d.lib
 - ❖ opencv_flann231d.lib
 - ❖ opencv_gpu231d.lib
 - ❖ opencv_haartraining_engined.lib
 - ❖ opencv_highgui231d.lib

- ❖ opencv_imgproc231d.lib
- ❖ opencv_legacy231d.lib
- ❖ opencv_ml231d.lib
- ❖ opencv_objdetect231d.lib
- ❖ opencv_ts231d.lib

Comme sur la Figure I.3

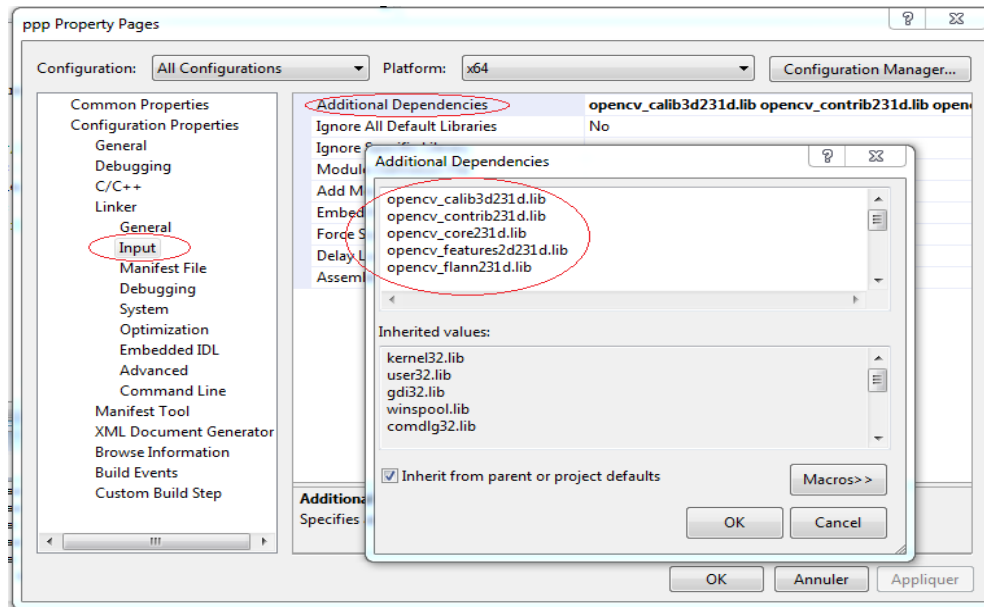


Figure I. 3 : Configuration nécessaire de Microsoft Visual Studio 2008 pour OpenCV

5. Conclusion

OPENCV offre gratuitement des outils de programmation pour gérer des données en format image ou des fichiers vidéo en entrée. Il contient de nombreuses fonctions prêtes à l'utilisation pour une large gamme d'applications.

OPENCV est un outil d'apprentissage très utilisé dans le domaine de la vision par ordinateur, il est gratuit et open source. Avec cet outil nous pouvons rapidement créer des applications qui permettent aux ordinateurs de voir et de prendre une décision basée sur ces données.

Chapitre II

Généralités sur les images et les Vidéos

1. Introduction

Le traitement des images numériques et des vidéos est une discipline nouvelle qui s'est développée rapidement grâce à l'émergence des nouvelles technologies de l'information qui permet de modifier une image ou une vidéo dans le but d'extraire des informations ou d'améliorer qualitativement ou quantitativement avec le codage de la compression des grand volumes d'informations qui s'impose alors comme une étape incontournable pour optimiser la taille mémoire à leurs archivages ainsi que le temps nécessaire à leur transmission à distance via les réseaux informatiques.

2. Définition d'image

Une image est la représentation d'une scène acquise à l'aide de systèmes de production d'images (appareils photographique, caméra, radiographies, scanner, sonar,...). Sa forme peut être analogique (ex: négatif, photographie, vidéo..) ou numérique (images numérisées suivant divers formats (images compressées ou non...) ou obtenues par des capteurs fournissant des images numérisées) et dans ce cas un traitement par ordinateur est possible [2].

3. Image numérique

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits). L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter [31].

4. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

4.1. Pixel «Picture Element»

Une image numérique est constituée d'un ensemble de points appelés pixels (pixel est une abréviation de (PICture ELeMent)) Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image [32].

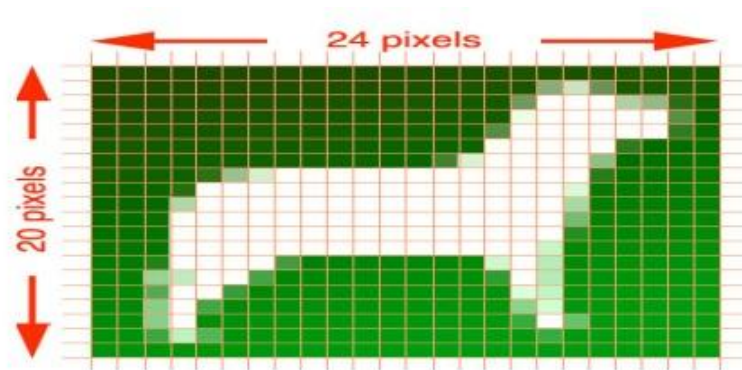


Figure II. 1: L'image comme un groupe de pixels [33].

Dans le cas d'une image monochrome, chaque pixel est codé sur 8 bits, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image. Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets: un octet pour chacune des couleurs: rouge (R), vert (V) et bleu (B). [34]

4.2. La définition et la résolution d'une image

On appelle la définition d'une image, le nombre de points (pixel) constituant l'image, c'est-à-dire sa «dimension informatique» (le nombre de colonnes de l'image que multiplie son nombre de lignes). Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480.

La résolution, terme souvent confondu avec la "définition", détermine par contre le nombre de points par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch); un pouce représentant 2.54 cm. La résolution permet ainsi d'établir le rapport entre le nombre de pixels d'une image et la taille réelle de sa représentation sur un support physique. Une résolution de 300 dpi signifie donc 300 colonnes et 300 rangées de pixels sur un pouce carré ce qui donne donc 90000 pixels sur un pouce carré. La résolution de référence de 72 dpi nous donne un pixel de 1"/72

(un pouce divisé par 72) soit 0.353mm, correspondant à un point pica (unité typographique anglo saxonne). [35]

4.3. Histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).

Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées. Prenons par exemple l'image suivante composée de niveaux de gris:

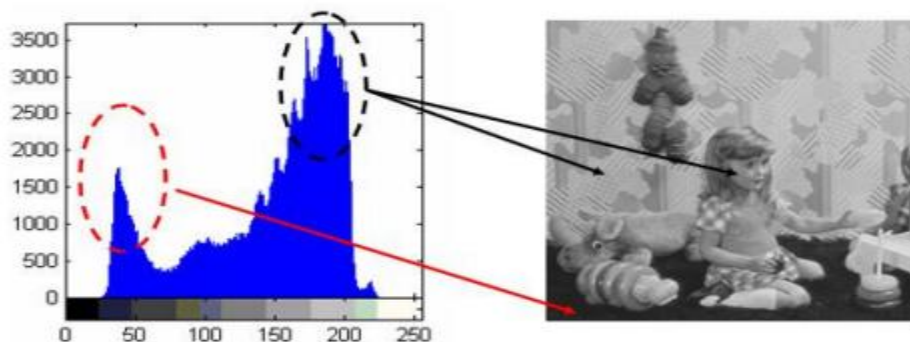


Figure II. 2: Histogramme obtenu pour l'image [36].

Pour les images en couleur plusieurs histogrammes sont nécessaires.

Par exemple pour une image codée en RGB:

- Un histogramme représentant la distribution de la luminance.
- Trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouges, bleues et vertes [37].

4.4. Contours et textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes [31].

4.5. 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 apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. La moyenne ou luminance (brillance) d'une image numérique en niveau de gris est définie comme la moyenne des pixels de l'image: [38]

$$Lum(I) = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

➤ Avec :

M: nombre de colonnes.

N: nombre de lignes.

$f(x, y)$: la valeur de niveaux de gris dans le pointe (x, y) .

4.6. 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 $L1$ et $L2$ sont les degrés de luminosité respectivement de deux zones voisines $A1$ et $A2$ d'une image, le contraste C est défini par le rapport:[31]

$$C = \frac{L1 - L2}{L1 + L2}$$

4.7. Profondeur

C'est le nombre de bits par pixel, cette valeur reflète le nombre de couleurs ou de niveaux de gris d'une image, par exemple :

- 32 bits/pixel = 1,07 milliards de couleurs.
- 24 bits = 16,7 millions de couleurs.
- 16 bits = 65 536 couleurs.
- 8 bits = 256 couleurs.

4.8. Le poids de l'image

Le poids d'une image se détermine en fonction de ces deux paramètres: dimensions, profondeur. Le poids de l'image est alors égal à sa dimension multipliée par sa profondeur. Par exemple, pour une image 640x480 en vraies couleurs (True colors) :

- Nombre de pixels (dimension) : $640 \times 480 = 307200$.
- Poids de chaque pixel (profondeur) : 24 bits = 3 octets.
- Le poids de l'image est ainsi égal à : $307200 \times 3 = 921600$ octets.

4.9. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. [34]

5. Représentation des couleurs

Il existe différentes catégories d'image selon le nombre de bit sur lequel est codée la valeur de chaque pixel.

5.1. L'image monochrome (binaire)

C'est le plus simple type d'image où chaque pixel peut prendre uniquement la valeur noire ou blanche (chaque pixel est codé sur un seul bit). C'est typiquement le type d'image que l'on utilise pour scanner du texte quand celui-ci est composé d'une seule couleur.

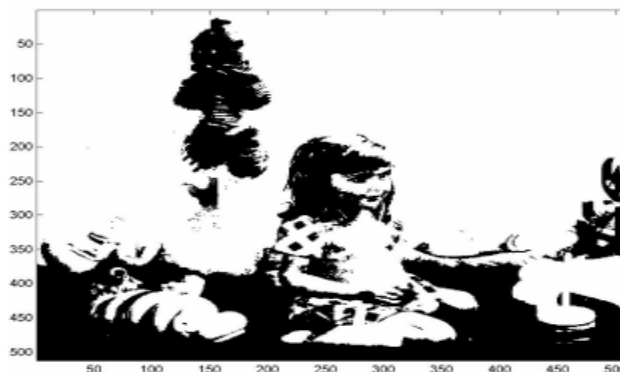


Figure II. 3: Une image binaire [36].

5.2. L'image en niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse à 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 (Figure II.4), on peut attribuer à chaque pixel de l'image une valeur correspondante 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.



Figure II. 4: Image en niveaux de gris [39].

5.3. L'image en couleurs

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc ou en niveau de gris, les applications multimédias utilisent le plus souvent des images en couleurs (Figure II.5).

La représentation des couleurs s'effectue de la même manière que les images en niveaux de gris avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation [38].



Figure II. 5: Image en couleurs.

Il existe plusieurs modes de représentation de la couleur, le plus utilisé pour les images numériques est le mode RGB. Ce mode est basé sur la synthèse additive des couleurs, c'est à dire que le mélange des trois composantes (R, G, B) donne une couleur.

6. Les formats d'image

➤ **Définition**

Un format d'image est une représentation informatique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler [30].

Nom du format	Créateur	Points forts	Points faibles	Note
JPEG et JPEG2000	Joint Photographic Experts Group	Excellente compression	Compression destructrice	Spécialement conçu pour les photographies, il est cependant à utiliser avec délicatesse tant sa compression peut brouiller l'image. Le format JPEG2000, évolution du format original, peut être réglé pour compresser sans pertes.
GIF (Graphical Interchange Format)	H&R Block et CompuServe	Possibilité d'animation et de transparence, compression efficace	Limité à 256 couleurs	Très répandu sur le Web malgré ses faiblesses et un problème de droit sur son format de compression. À déconseiller pour les photos.

PNG (Portable Network Graphics)	W3C	Excellente compression sans perte. Possibilité de transparence. Standard donc pérenne.	Pas très efficace pour les larges photographies	Format destiné à remplacer le format GIF et ses limitations, mais ayant encore du mail à s'implanter dans les habitudes des développeurs. Peut remplacer les JPEG comme les GIF (sauf en ce qui concerne l'animation).
TIFF (Tagged Image File Format)	Aldus	Compression sans perte efficace. Couche de transparence	Lourdeur des fichiers non compressés. Format propriétaire.	Format de stockage très utilisé, à éviter pour le Web
BMP (Bitmap)	Microsoft	Format par défaut de Windows	Disponible uniquement sur la plateforme de Microsoft	Généralement non compressé le format RLE est possible mais peu reconnu
PSD (Photoshop Document)	Adobe	Très complet, conserve toutes les couches de l'image.	Taille prohibitive	Format standard de l'outil Photoshop, très populaire mais à limiter au stockage de retouche d'image du fait de la taille des fichiers

Tableau II.1: Les différents formats d'image [40].

7. Compression des images

7.1. L'objectif de la compression des images

L'objectif principal de la compression d'image est de réduire la quantité d'informations nécessaire à sa représentation. Cette réduction est obtenue en minimisant la redondance de l'information présente dans une image. Cette redondance peut être d'origine statistique, spatiale ou encore fréquentielle. Les méthodes de compression d'images se différencient par la technique utilisée pour minimiser cette redondance tout en sauvegardant les informations pertinentes présentes dans ces images. [3]

7.2. Caractérisation de la compression

La compression peut se définir par le quotient de compression, c'est-à-dire le quotient du nombre de bits dans l'image compressée par le nombre de bits dans l'image originale

➤ Taux de compression

Le taux de compression sert à évaluer la qualité de la compression en faisant l'opération suivante:

Taux compression = (taille-img-origine - taille-img-compressé)/(taille-img-origine)

7.3. Les types des méthodes de compressions des images

7.3.1. Compression physique et compression logique

La compression physique agit directement sur les données; il s'agit ainsi de regarder les données redondantes d'un train de bits à un autre. La compression logique par contre est effectuée par un raisonnement logique en substituant une information par une information équivalente [41].

7.3.2. Compression symétrique et asymétrique

Dans le cas de la compression symétrique, la même méthode est utilisée pour compresser et décompresser l'information, il faut donc la même quantité de travail pour chacune de ces opérations. C'est ce type de compression qui est généralement utilisée dans les transmissions de données [41].

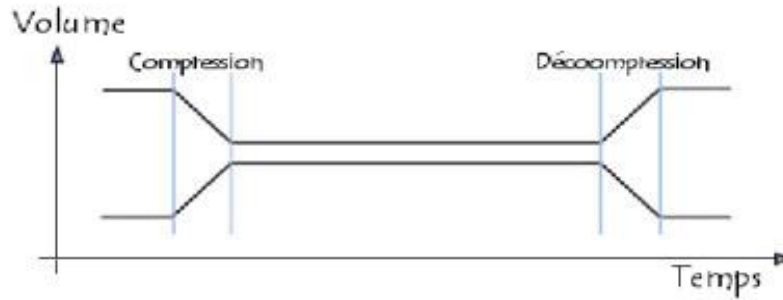


Figure II. 6: compression de type symétrique. [41]

La compression asymétrique demande plus de travail pour l'une des deux opérations, on recherche souvent des algorithmes pour lesquels la compression est plus lente que la décompression. Des algorithmes plus rapides en compression qu'en décompression peuvent être nécessaires lorsque l'on archive des données auxquelles on n'accède peu souvent (pour des raisons de sécurité par exemple), car cela crée des fichiers compacts [41].

7.3.3. Compression adaptif, semi adaptif et non adaptif

Certains algorithmes de compression sont basés sur des dictionnaires spécifiques à un type de données : ce sont des encodeurs non adaptifs. Les occurrences de lettres dans un fichier texte par exemple dépendent de la langue dans laquelle celui-ci est écrit. Un encodeur adaptif s'adapte aux données qu'il va devoir compresser, il ne part pas avec un dictionnaire déjà préparé pour un type de données. Enfin un encodeur semi-adaptif construira celui-ci en fonction des données à compresser : il construit le dictionnaire en parcourant le fichier, puis compresses ce dernier [41].

7.3.4. Compression sans pertes et la compression avec pertes :

➤ *La compression sans pertes*

Les méthodes de compression sans pertes, comme leur nom l'indique aucune perte d'informations, si l'image a été compressée avec une méthode sans pertes, l'image originale peut exactement se restituer à partir de l'image compressée, la compression sans pertes est généralement utilisée pour des applications qui ne tolèrent aucune différence entre l'image originale et l'image reconstruite. Le taux de compression sans

pertes est généralement autour de 2:1 ce qui est insuffisant pour le stockage ou pour la transmission [4].

Parmi les méthodes de compression sans perte, on trouve la méthode RLE, la méthode HUFFMAN...etc.

❖ Méthode de codage RLE :

La méthode de compression RLE (Run Length Encoding) est utilisée par de nombreux formats d'images. Elle est basée sur la répétition d'éléments consécutifs. [5]

Le principe de base consiste à coder un premier élément donnant le nombre de répétitions d'une valeur puis le compléter par la valeur à répéter. Ainsi selon ce principe la chaîne "DDDDDDBBBBBBBBBBBB" compressée donne "6D13B". Le gain de compression est ainsi de $(19-5)/19$ soit environ 73,7%. En contrepartie pour la chaîne "REELLEMENT", dans lequel la redondance des caractères est faible, le résultat de la compression donne "1R2E2L1E1M1E1N1T"; la compression s'avère ici très coûteuse, avec un gain négatif valant $(10-16)/10$ soit -60%.

❖ Le codage Huffman

Créer des codes à longueurs variables sur un nombre entier de bits. L'algorithme considère chaque message à coder comme étant une feuille d'un arbre qui reste à construire [6]. L'idée est d'attribuer aux messages de plus faibles probabilités, les mots codés les plus longs et les mots codés les plus courts pour les messages de fortes probabilités.

➤ *La compression avec pertes*

La compression avec pertes (en anglais lossy compression), par opposition à la compression sans pertes, se permet d'éliminer quelques informations pour avoir le meilleur taux de compression possible, tout en gardant un résultat qui soit le plus proche possible des données originales. C'est le cas par exemple de certaines compressions d'images ou de sons. Le taux de compression avec pertes est généralement beaucoup plus élevé à ceux obtenus par les méthodes de compression sans pertes.

8. Définition d'une vidéo numérique

Une vidéo est une succession d'images à une certaine cadence. L'œil humain a comme caractéristique d'être capable de distinguer environ 20 images par seconde. Ainsi, en affichant plus de 20 images par seconde, il est possible de tromper l'œil et de lui faire croire à une image animée. On caractérise la fluidité d'une vidéo par le nombre d'images par secondes (en anglais frame rate), exprimé en FPS (Frames per second, en français frames par seconde). D'autre part la vidéo au sens multimédia du terme est généralement accompagnée de son, c'est-à-dire de données audio [41].

9. Les formats vidéo

De la même manière qu'un fichier image, il existe plusieurs formats de fichiers vidéo, à l'intérieur des quels le choix de la définition (qualité) est inversement proportionnelle à la taille du fichier généré.

Un fichier vidéo c'est presque toujours vidéo + son [7].

9.1. Fichier Conteneur

Les vidéos numériques, depuis qu'elles sont manipulées par des ordinateurs, sont contenues dans des fichiers. Ces fichiers sont des conteneurs pour la vidéo.

Un fichier conteneur (wrapper ou container en anglais) est un format de fichier qui contient les flux vidéo et/ou audio et des métadonnées (titres, sous-titres, chapitres, menus, ...). Le nom du fichier est suivi d'une extension ".xxx". Le choix du conteneur a une influence sur son utilisation : certains ne contiennent que le flux audio/vidéo [52].

9.2. Le piège du pixel flou

Pour commencer, intéressons-nous ici à la forme du pixel de prise de vue (pixel télé) et du pixel d'affichage (pixel informatique).

C'est le plus gros piège de la vidéo digitale:

- Le pixel informatique est carré.
- Le pixel télé est rectangulaire vertical ou horizontal:
 - ❖ télé 4:3 PAL/SECAM : pixel rectangulaire étroit.
 - ❖ télé 4:3 NTSC : pixel rectangulaire long.
 - ❖ télé 16:9 (HD) : pixel rectangulaire très long. [7][8]

Le modèle ci-dessous est un modèle de pixel informatique est carré il donne une déformation de l'image et un flou global, très perceptible sur les textes (Figure II.7).



Figure II. 7: Modèle de pixel informatique est carré [53].

L'affichage sans déformation (576p = 1024x576) par le modèle de télé 16:9 (HD) : pixel rectangulaire très long donne ceci :



Figure II. 8 : Modèle de L'affichage sans déformation [53].

Remarque: Le présentateur à bien meilleure mine et le texte est net !

10. Les formats de fichiers vidéo

Comme mentionné, certains formats sont des "conteneurs", c'est-à dire qu'ils peuvent contenir un peu n'importe quoi (par ex: divers types de compression, de résolution, etc.)

Les formats les plus courants sont :

- **AVI (Audio Video Interleave)** : Format de fichier d'animations mis au point par Microsoft, qui fonctionne sur n'importe quelle machine, et ne nécessite pas de carte d'extension particulière. Dans ce format, la compression est toujours effectuée image par image.
- **MPEG (Motion Picture Expert Group)**:Format de fichier audio et vidéo utilisant la compression avec perte. Utilisé pour la diffusion en continu sur Internet (les streaming). Le principe est d'obtenir des fichiers légers en supprimant les parties d'images fixes lors du passage d'une image à une autre image [43]. Il existe plusieurs standards de MPEG comme le MPEG-1, le MPEG-2, le MPEG-4, et MPEG-7.
 - ❖ **MPEG-1**, développé en 1988, est un standard pour la compression des données vidéo et des canaux audio associés (jusqu'à 2 canaux pour une écoute stéréo). Il permet le stockage de vidéos à un débit de 1.5Mbps dans une qualité proche des cassettes VHS sur un support CD appelé VCD (*Vidéo CD*).
 - ❖ **MPEG-2**, un standard dédié originalement à la télévision numérique (*HDTV*) offrant une qualité élevée à un débit pouvant aller jusqu'à 40 Mbps, et 5 canaux audio sur surround. Le MPEG-2 permet de plus une identification et une protection contre le piratage. Il s'agit du format utilisé par les DVD vidéo.
 - ❖ **MPEG-4**, un standard destiné à permettre le codage de données multimédia sous formes d'objets numériques, afin d'obtenir une plus grande interactivité, ce qui rend son usage particulièrement adapté au Web et aux périphériques mobiles.

- ❖ **MPEG-7**, un standard visant à fournir une représentation standard des données audio et visuelles afin de rendre possible la recherche d'information dans de tels flux de données. Ce standard est ainsi également intitulé MultiMedia Content Description Interface [42].
- **wmv (Windows Media Video)** : Ces fichiers sont obtenus, par exemple, lors de la création d'un diaporama avec Photorecité ou Windows Movie Maker. C'est un format propriétaire de Microsoft lisible sur PC mais pas sur platine de salon.
- **VOB (Video Object file)** : Utilisé pour créer des applications sur DVD. Les fichiers VOB sont lisibles par les platines de salon.
- **Mov** : Ce format désigne un type de fichiers vidéos au standard Quicktime et donc lisibles avec le lecteur QuickTime. Ce format est moins courant que le format *.avi. Il est utilisé par quelques marques d'appareils photo numériques comme Sony, pour créer des séquences vidéo.
- **mp4** : Ce format (format vidéo compressé) est très utilisé sur Internet. Il permet l'affichage d'un flux vidéo à mesure qu'il est diffusé. Pour lire une vidéo enregistrée dans le format MP4, il suffit d'utiliser un lecteur multimédia comme le lecteur VLC.
- **dv (ou .mov avec compression DV)** : idéal pour le montage vidéo mais surtout pas pour la diffusion.
- **Flv (Flash Video)** : pas mal utilisé sur le web car les vidéos en ligne ne sont pas/pas facilement copiables. A proscrire donc si on veut que le lecteur puisse aisément récupérer la vidéo. Ne pas oublier que c'est un format propriétaire, donc peu d'outils simples et/ou gratuits pour le manipuler [43].

11.Représentation d'une séquence vidéo

Une séquence vidéo brute est une suite d'images fixes, qui peut être caractérisée par trois principaux paramètres : sa Résolution en luminance, sa Résolution spatiale et sa Résolution temporelle.

11.1. La résolution en luminance

Détermine le nombre de nuances ou de couleurs possibles pour un pixel. Celle-ci est généralement de 8 bits pour les niveaux de gris et de 24 bits pour les séquences en couleurs.

11.2. La résolution spatiale

Définit le nombre de lignes et de colonnes de la matrice de pixels.

11.3. La résolution temporelle

Est le nombre d'images par seconde. La valeur de ces trois paramètres détermine l'espace mémoire nécessaire pour stocker chaque image de la séquence. Cet espace mémoire est caractérisé par le débit, qui est le coût de stockage pour une seconde (capacité mémoire nécessaire pour stocker une seconde de vidéo) [9].

Par exemple :

Une séquence ayant une résolution de 720 par 576 pixels, un codage des couleurs sur 24 bits, et une fréquence de 25 images par seconde, nécessitera un débit de 137Mb/s. Le débit d'une séquence vidéo brute est très élevé comparé aux débits et à l'espace offert par les moyens de stockage et de transferts actuels.

12. Compression de la vidéo numérique

Les technologies de compression vidéo ont pour but de réduire et de supprimer les données vidéo redondantes de façon à permettre la transmission et le stockage efficaces d'un fichier vidéo numérique sur des disques d'ordinateur par le biais d'un réseau. Avec des techniques de compression performantes, il est possible d'obtenir une réduction sensible de la taille de fichier avec un effet minimal, voire nul, sur la qualité visuelle. La qualité vidéo, toutefois, peut être affectée si la taille de fichier est réduite davantage en augmentant le niveau de compression pour une technique de compression donnée [44].

12.1. Codec

Le processus de compression consiste à traiter la source vidéo à l'aide d'un algorithme afin de créer un fichier compressé prêt pour la transmission ou le stockage. Pour lire le fichier compressé, un algorithme inverse est appliqué, ce qui permet d'obtenir une vidéo contenant pratiquement le même contenu que la source vidéo d'origine. Le temps pris pour la compression, l'envoi, la décompression et l'affichage d'un fichier est ce que l'on appelle la latence. Plus l'algorithme de compression est perfectionné, plus la latence est élevée.

Une paire d'algorithmes travaillant ensemble est ce que l'on appelle un codec (encodeur/décodeur) vidéo. Les codecs vidéo de différentes normes ne sont normalement pas compatibles les uns avec les autres. En d'autres termes, le contenu vidéo compressé à l'aide d'un algorithme ne peut pas être décompressé à l'aide d'un algorithme différent. Par exemple, un décodeur MPEG-4 n'est pas compatible avec un encodeur H.264. L'explication réside dans le fait qu'un algorithme ne peut pas décoder correctement le résultat obtenu par un autre algorithme, mais il est possible d'utiliser plusieurs algorithmes différents dans le même logiciel ou matériel, ce qui autorise la coexistence de plusieurs formats [44].

12.2. Algorithmes de compression Vidéo

Les normes de compression n'utilisent pas toutes les mêmes méthodes pour réduire les données, ce qui explique que les résultats diffèrent en termes de débit, de qualité et de latence. Il existe deux types d'algorithmes de compression : les algorithmes de compression d'images et les algorithmes de compression vidéo.

12.2.1. La compression d'images (Technique intra-image)

Les algorithmes de compression vidéo intra-image s'appuient sur les principes classiques de compression d'image où chaque image est travaillée individuellement. Les données sont réduites en supprimant les informations redondantes ou inutiles car non remarquables par l'œil humain [45]. Il existe plusieurs techniques. En voici quelques exemples, sans rentrer dans les détails mathématiques.

- **Transformation des couleurs** : le système de stockage colorimétrique est modifié par un système moins gourmand en taille mémoire[44]. Par exemple, le codec MPEG (1, 2 ou 4) transforme le classique système RVB (image de gauche sur Figure II.9) qui enregistre pour chaque pixel la valeur du rouge, du vert et du bleu, en un système plus proche de la représentation de l'œil humain et sollicitant moins d'espace de stockage, le système YUV intégrant une composante de luminosité et deux valeurs de chrominance (image de droite sur figure II.9).

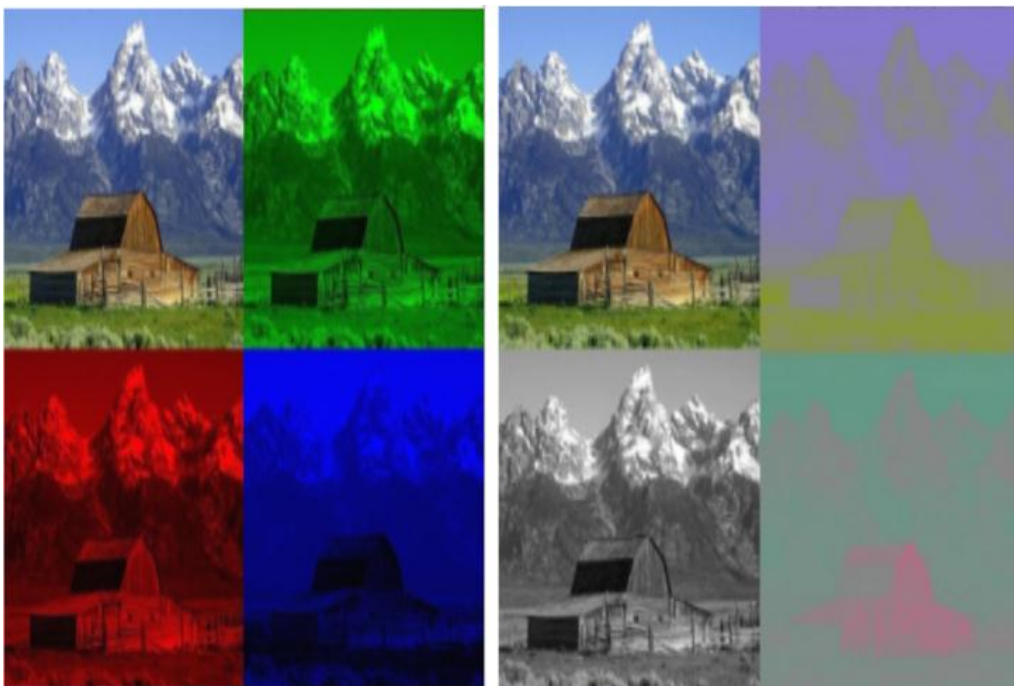


Figure II. 9: Exemple de technique intra-image par transformation des couleurs. [54]

- **Élimination des redondances** : Dans une image, des blocs peuvent se reproduire à l'identique ou, éventuellement, avec de très légères variantes. Le premier bloc d'une série étant codé, il devient redondant de coder les autres blocs qui lui sont identiques, ou qui disposent d'un composant colorimétrique identique. Il suffit d'indiquer qu'il y a répétition du premier bloc. La compression spatiale identifie les zones redondantes au sein d'une même image et ne stocke que les différences.



Figure II. 10: Exemple de technique intra-image par Elimination les redondances. [44]

12.2.2. La compression vidéo (Technique inter-image)

En examinant une vidéo, il est facile de remarquer qu’il existe dans certaines séquences des éléments redondants fixes ou mobiles mais suivant un mouvement logique. A l’échelle du pixel, on parle donc de corrélation temporelle entre des images successives. Dès lors, une image est décrite en fonction de sa différence par rapport à la précédente. Les zones redondantes sont reprises des images précédentes. Cela implique différents mécanismes d’identification des redondances [10].

- **Le codage différentiel** : si le premier mécanisme, où une image est comparée à une image de référence et seuls les pixels qui ont changé par rapport à cette image de référence sont codés. Cela permet de réduire le nombre de valeurs codées et stockées pour chaque pixel. Le codage différentiel compare des éléments statiques d’une image à l’autre [46].

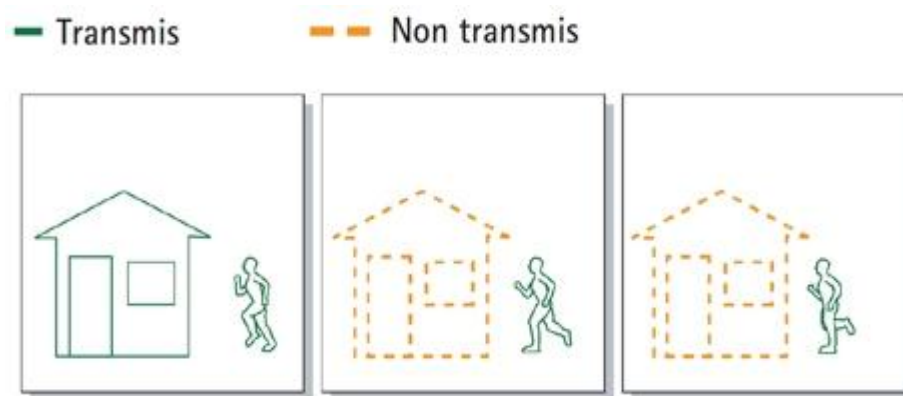


Figure II. 11: Exemple sur la technique Inter-image par le codage différentiel [44].

- **La compensation de mouvement** : Est plus complexe, utilisée afin de réduire encore davantage la quantité de données. Cette technique tient compte du fait qu'une bonne partie de ce qui compose une nouvelle image dans une séquence vidéo se retrouve dans une image adjacente, mais pas forcément au même endroit. Il est donc possible de composer ou de « prédire » une nouvelle image bloc par bloc en recherchant un bloc identique dans une image de référence. Si un bloc référent est trouvé, l'encodeur code l'emplacement dans l'image de référence où se situe le bloc identique ainsi que le vecteur de mouvement qui utilise moins d'espace que le codage du contenu réel d'un bloc [46].

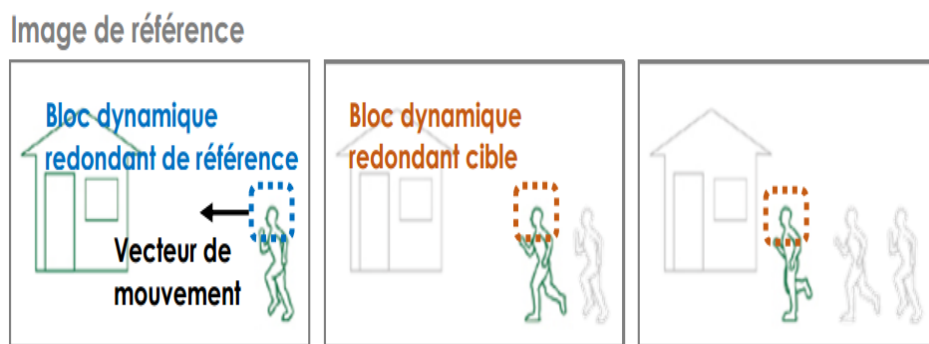


Figure II. 12: Exemple sur la technique Inter-image par La compensation de mouvement [44].

Avec la prédiction inter-images, chaque image d'une séquence est classifiée comme un certain type d'image (image I, image P ou image B).

- ❖ **Une image I**, ou intra-image, est une image autonome qui peut être décodée indépendamment, sans référence à d'autres images. La première image d'une séquence vidéo est toujours une intra-image. Elle fait généralement l'objet d'une compression spatiale intra-image. L'intégration régulière d'image I garantit la qualité résultant de la décompression.
- ❖ **Une image P** est une inter-image prédictive qui fait référence aux parties des images I et/ou P antérieures pour le codage de l'image. Les images P nécessitent généralement moins de bits que les images I, mais elles sont généralement très sensibles aux erreurs de transmission en raison de la dépendance complexe vis-à-vis des images de référence P et I antérieures.

- ❖ Une image *B* est une inter-image bi-prédictive qui fait référence à une image antérieure ainsi qu'à une image future [45].

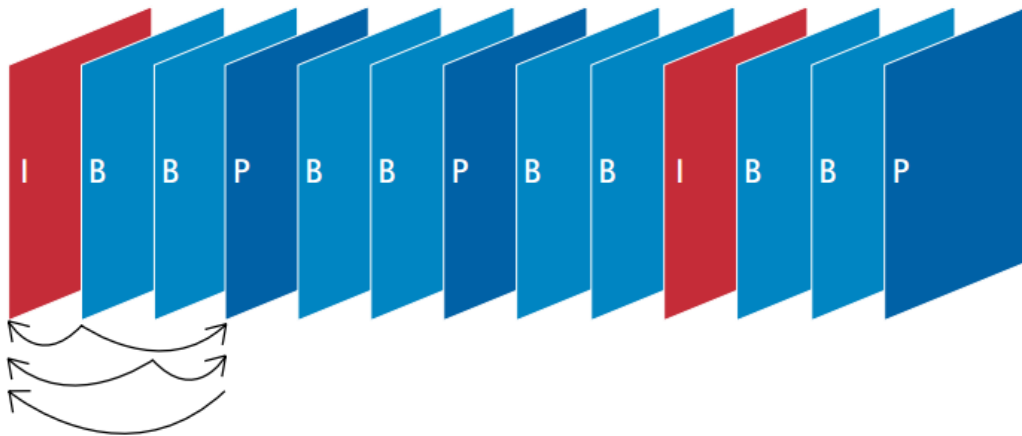


Figure II. 13: Séquence type avec des images I, B et P [9].

Lorsqu'un décodeur vidéo rétablit une vidéo en décodant le train de bits image par image, le décodage doit toujours commencer par une image I. Les images P et B, si elles sont utilisées, doivent être décodées avec les images de référence.

13. Conclusion

Dans ce chapitre, nous avons présentés une étude générale sur les caractéristiques fondamentales des images et des vidéos dans le but de les améliorer ou d'extraire des informations pertinents. Donc, les objectifs principale de la compression est de réduire la taille des images et des vidéos.

Chapitre III

Détection des visages

1. Introduction

La détection des visages et des composantes faciales a connu un grand intérêt récemment. C'est la première étape dans n'importe quel système de traitement de visage tel que la reconnaissance du visage, le brouillage du visage et l'escamotage des yeux.

L'objectif de ce travail est de brouiller les visages ou de cacher les yeux, Mais avant cela il est indispensable de localiser le visage, et vue l'importance de ce domaine, les recherches ont donné plusieurs méthodes ayant des principes différents.

2. Définition de détection des visages

La détection de visage est un domaine de la vision par ordinateur consistant à détecter un visage humain dans une image numérique. C'est un cas spécifique de détection d'objet, où l'on cherche à détecter la présence et la localisation précise d'un ou plusieurs visages dans une image. C'est l'un des domaines de la vision par ordinateur parmi les plus étudiés, avec de très nombreuses publications, brevets, et de conférences spécialisées. La forte activité de recherche en détection de visage a également permis de faire émerger des méthodes génériques de détection d'objet [47].

La détection de la présence ou non d'un visage dans une image soit basée sur des éléments stables et relativement descriptifs du visage humain et qui permettent ensuite de le reconnaître. Parmi ces éléments, on peut citer la forme du visage, la couleur de la peau, le contour des yeux, la forme du nez ou de la bouche...



Figure III. 1: Exemple d'images comportant un seul visage [48]

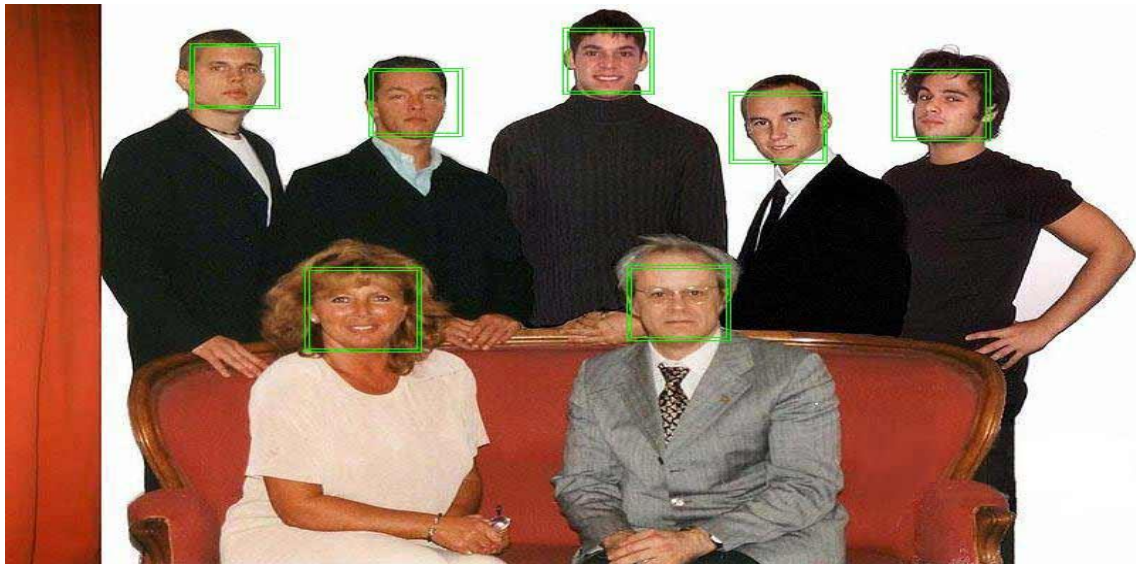


Figure III. 2: Exemple d'images comportant plusieurs visages [47]

2.1. Application

La détection de visage a de très nombreuses applications directes en vidéo-surveillance, biométrie, robotique, commande d'interface homme-machine, photographie, indexation d'images et de vidéos, brouillage du visage, etc..... Elle permet également de faciliter l'automatisation complète d'autres processus comme la reconnaissance de visage ou la reconnaissance d'expressions faciales.

Parmi les applications directes, la plus connue est sa présence dans de nombreux appareils photo numérique, où elle sert à effectuer la mise au point automatique sur les visages. C'est également une technique importante pour les interfaces homme-machine évoluées, afin de permettre une interaction plus naturelle entre un humain et un ordinateur.

La détection de visage est aussi utilisée en indexation d'images et recherche d'information, où elle peut être utilisée pour rechercher des images contenant des personnes, associer automatiquement un visage à un nom dans une page web, identifier les principales personnes dans une vidéo par clustering.

La détection de visage peut aussi servir à déterminer l'attention d'un utilisateur, par exemple face à un écran dans l'espace public, qui peut également, une fois le visage détecté déterminer le sexe et l'âge de la personne afin de proposer des publicités ciblées.

Cela peut également servir à savoir si une personne est bien présente devant une télévision allumée, et dans le cas contraire mettre l'appareil en veille ou réduire la luminosité afin d'économiser de l'énergie.

3. Les caractéristiques du visage

3.1. Détection des yeux

La détection des yeux, ou des centres des yeux, est l'une des étapes les plus importantes dans le processus de reconnaissance. En effet, la taille du visage est définie par la distance intraoculaire, l'orientation par l'alignement des yeux. Puisque la détection des yeux va permettre de normaliser ensuite les images ce qui est indispensable avant tout algorithme de mise en correspondance. Pour cela, on trouve dans la littérature plusieurs méthodes de détection des yeux. De l'efficacité de ces méthodes dépend la performance de l'algorithme de reconnaissance. Plusieurs approches ont été utilisées pour détecter la position des yeux. Ces approches peuvent être classées en trois catégories [10] [11].

- **Approche basée sur l'apparence** : Cette méthode détecte les yeux en se basant sur l'intensité lumineuse de l'œil. [12] On utilise la projection du niveau de gris pour détecter les yeux.
- **Approche basée sur les modèles**: Cette approche construit un modèle (Template) déformable de l'œil qui est mis en comparaison avec l'œil de l'image de test.
- **Approche basée sur les traits de l'œil** : On s'intéresse dans cette méthode aux traits caractéristiques des yeux tels que les coins des yeux [13]. L'algorithme de détection se base sur deux filtrages : un filtrage pour la détection de l'œil et un second pour la détection des coins des yeux.

3.2. Détection des bouches

L'un des éléments du visage ayant la forme la plus simple, en l'absence d'expression, est la bouche. Elle est toujours présentée sous forme d'une ligne dans l'image de contour. Nous allons chercher une ligne qui se trouve dans une des zones de présélection [14].

Il existe plusieurs modèles pour détecter la bouche. La plus connue est la transformée de Hough. Elle est devenue durant ces dernières décennies la méthode la plus utilisée dans le domaine de la vision artificielle.

3.3. Détection de nez

Le nez est un organe qui se caractérise par sa forme fixe par rapport aux mouvements de la tête. Les petites rotations et les expressions faciales ne changent pas la forme du nez. Cela fait du centre du nez un point stable et facilement détectable. Il est utilisé principalement dans la reconnaissance et la segmentation du visage en 3D ainsi que dans le suivi du visage dans un flux vidéo [14].

4. Les méthodes de détection des visages

Depuis quelques années la détection des formes, des objets ainsi que celle des visages sont prises comme domaines de recherche par plusieurs personnes et sociétés ce qui amène à l'existence de plusieurs approches pour la détection des visages dans une image [15], parmi ces approches on cite les suivantes :

4.1. Méthode basée sur les caractéristiques invariables (Feature Invariant)

Ces approches sont utilisées principalement pour la localisation de visage .les algorithmes développés visent à trouver les caractéristiques structurales existantes même si la pose, le point de vue ou la condition d'éclairage changent. Puis ils emploient ces caractéristiques invariables pour localiser les visages.

Le problème avec cette méthode est que la qualité des images peut être sévèrement diminuée à cause de l'illumination, le bruit ou l'occlusion ce qui empêche l'algorithme de fonctionner correctement.

Nous pouvons citer trois familles des méthodes appartenant à cette approche qu'on détaillera ci-dessous.

4.1.1. Caractéristiques du visage (Facial Feature)

Cette méthode utilise les plans d'arrêtes appelés "Canny detector" et des heuristiques pour supprimer tous les groupes d'arrêtes sauf celles qui représentent les contours du visage. Une ellipse est déduite comme frontière entre l'arrière-plan et le visage. Celle-ci est décrite comme étant formée des points de discontinuité dans la fonction de luminance (intensité) de l'image. Le principe de base consiste à reconnaître des objets dans une image à partir de modèles de contours connus aux préalables. Pour réaliser cette tâche, deux méthodes seront présentées : la transformée de Hough et la distance de Hausdorff.

- **Distance de Hausdorff** : Cette méthode utilise quant à elle les arêtes comme données de base ainsi qu'un algorithme spécial de 'Template Matching'. En effet, la distance de Hausdorff vise à mesurer la distance entre deux ensembles de points séparés, qui sont la plupart du temps une carte d'arêtes (image de recherche) et un modèle prédéfini.

L'algorithme de base effectue la recherche des meilleurs endroits de correspondance partout dans l'image (translation) et aussi pour différentes rotations. Cette recherche peut également inclure un facteur d'échelle afin de détecter des variations du modèle original. Cette méthode a été utilisée avec succès pour la détection des visages qui présentent des vues frontales (faciale) [16].

- **Transformée de Hough**: La transformée de Hough est une méthode permettant d'extraire et de localiser des groupes de points respectant certaines caractéristiques, équation d'une forme bien déterminée.

L'algorithme de base a également été modifié pour voir ainsi apparaître plusieurs variantes, dont la "Randomized Hough Transform" (la transformée de Hough aléatoire), qui peut être appliquée à la recherche des formes quelconques tout comme des cercles ou des ellipses (exp : Visages).

Finalement, la transformée de Hough peut être utilisée pour détecter les yeux et la bouche. Par contre, cette méthode échouera lorsque l'image est trop petite ou lorsque les yeux ne sont pas clairement visibles [49].

4.1.2. Texture

La texture de l'être humain est distinctive et peut être utilisée pour séparer les visages par rapport à d'autres objets. « Augusteijn et Skufca » ont développé une méthode de détection de visages sur une image en se basant uniquement sur la texture. [16]

Dans cette méthode trois types de caractéristiques sont pris en considération : la peau, les cheveux et le reste des composants de visage.

4.1.3. Couleur de la peau (Skin Color)

Pour les visages, la couleur de la peau de l'être humain est une caractéristique spécifique c'est pour cela qu'elle est utilisée dans plusieurs applications telles que la détection du visage, la segmentation vidéo, et le brouillage du visage. En effet la couleur de la peau est différente selon les personnes et leurs origines (Africains, Européennes, Asiatiques...). Dans ce contexte plusieurs études ont démontré que la plus grande différence s'étend largement entre l'intensité plutôt que la chrominance lumineuse [17].

➤ *Espace de couleur*

La première étape pour détecter la peau est de choisir un espace de couleur adéquat, le choix d'un espace particulier dépend souvent de la méthode envisagée et du résultat recherché. L'espace de couleur étant souvent utilisé pour coder ou bien brouiller les images

Nous pouvons utiliser plusieurs types d'espaces de couleurs (RGB, RGB normalisé, HSV, YcrCb, YIQ, YES, CIE XYZ, CIE LUV, etc.)

L'un des espaces le plus couramment utilisé pour effectuer cette tâche est le HSV. Son principal avantage réside dans le fait qu'une de ses composantes représente la luminosité (le V de Value), la teinte (le H de Hue) et la saturation (le S de Saturation).

- La teinte (en anglais Hue), correspondant à la perception de la couleur, et elle est codée suivant l'angle qui lui correspond sur le cercle des couleurs: 0° ou 360° : rouge ; 60° : jaune ; 120° : vert ; 180° : cyan ; 240° : bleu ; 300° : magenta. [30]

- La saturation, décrivant la pureté de la couleur, c'est-à-dire son caractère vif ou terne. Elle varie entre 0 et 100 %. [30]
- La luminance (en anglais value), indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre. Elle varie entre 0 et 100 %. [30]

La saturation et la teinte caractérisent la chromaticité du signal : cela est très proche du système perceptif de notre corps. La brillance est une grandeur séparée, ce qui permet de faire des considérations sur les dégradés d'intensité indépendamment des considérations de couleurs [18]. La figure III.3 représente cette espace de couleur.

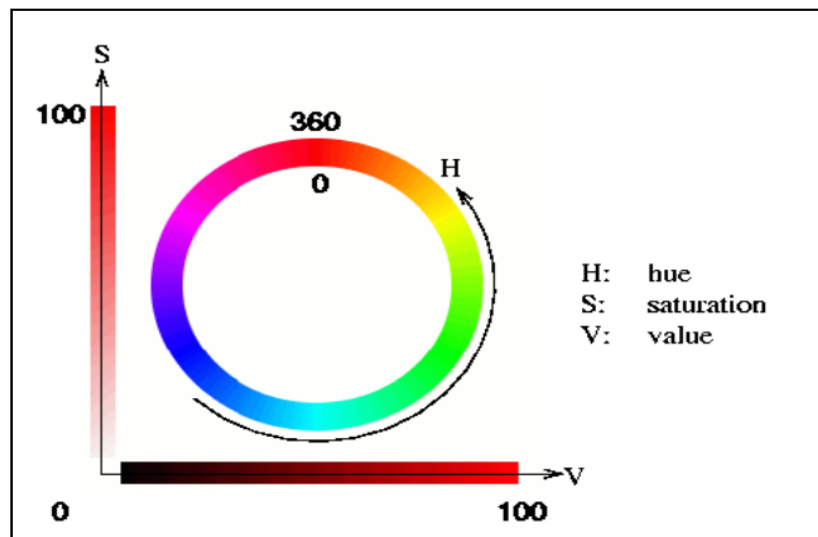


Figure III. 3: Système de couleur HSV [19].

La figure suivante représente une image originale (image droite) et une image convertie à l'espace de couleur HSV (image gauche).

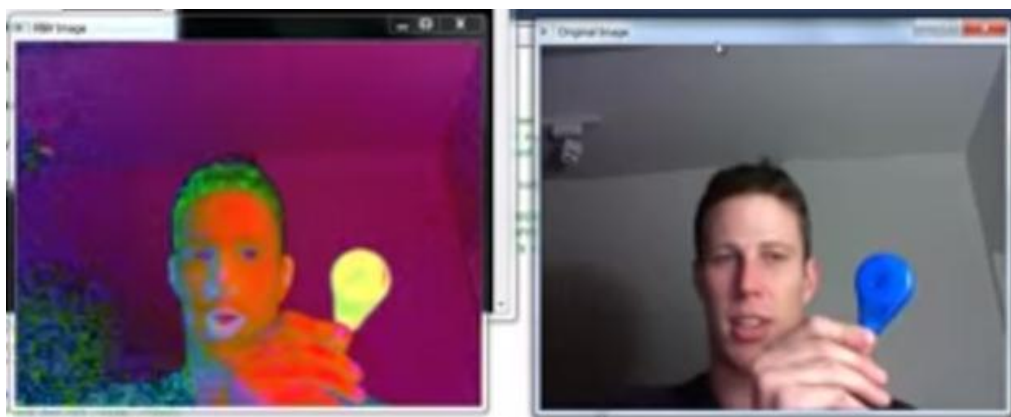


Figure III. 4: Image convertie à l'espace de couleurs HSV [50].

Nous remarquons sur l'image gauche de la figure III.4 qui est représentée avec l'espace HSV qu'elle n'est pas lisible et cela empêche l'identification de la personne. Et pour cette raison, nous avons utilisé cette technique dans notre application afin de flouter le visage.

4.2. Approche basée sur la reconnaissance

C'est une méthode fondée sur des règles qui représentent les composants principaux et représentatifs des visages humains. Les règles sont généralement constituées à partir de la relation entre les caractéristiques du visage. Par exemple, les visages sur les images ont souvent deux yeux qui sont symétriques, un nez et une bouche. La relation entre ces membres peut être représentée par la distance entre ces derniers et leur position.

Un problème se pose avec cette approche, en l'occurrence la difficulté de traduire les connaissances relatives aux visages humains en règle bien définies, ce qui peut provoquer des erreurs de détection et rendre le système peu fiable[55].

4.3. Appariement de gabarits (Template Matching)

La détection de visage entiers ou de parties de visage se fait à travers un apprentissage d'exemples standards de visages. La corrélation entre les images d'entrées et les exemples enregistrés est calculée et utilisée pour la décision.

On trouve deux types de corrélation suivant le type des gabarits.

4.3.1 Des faces prédéfinies de visages

Cette technique est utilisée pour classer des objets, elle est très intéressante pour la détection de visage de par sa facilité d'application.

Le principe de cette méthode basé sur une comparaison effectuée entre une image quelconque et un modèle prédéfini, dont le but est de calculer la corrélation pour aboutir à une décision par oui ou non. La correspondance est faite pixel par pixel [55].

4.3.2 Modèle déformable

Cette approche est utilisée dans le but de modéliser les caractéristiques faciales qui s'adaptent élastiquement par rapport au modèle du visage présent. Dans cette méthode, les caractéristiques faciales sont décrites par des gabarits paramètres. Une fonction est définie pour relier les contours, les sommets et les angles dans l'image d'entrée, pour faire correspondre les paramètres sur les gabarits. La meilleure adaptation du modèle élastique est de trouver la fonction énergétique en minimisant les paramètres. Cette méthode a été utilisée par Yuille et Al [16].

4.4. Méthode basée sur les apparences (Appearance-based method)

La différence entre cette méthode et " Template Matching " est que les modèles(Template) sont lus à partir des images d'apprentissage qui doivent être représentatives et faites à différentes positions du visage [49].

Généralement "appearance-based method" se base sur des techniques d'analyse statistiques (pourcentage d'existence des modèles dans l'image) et d'apprentissage automatique pour trouver des caractéristiques significatives des visages et des non visages.

Pour l'approche suivante il existe plusieurs méthodes où chacune d'entre elles se base sur une des caractéristiques du visage ou plus précisément une partie de visage qui peut être interprétée sous le cadre probabiliste.

On peut citer parmi celles-ci, la méthode basée sur les réseaux de neurones de Rowley et al. La méthode de Schneiderman et Kanade basée sur un classifieur de Bayes naïf ainsi que le fameux algorithme de Viola et Jones fonctionnant en temps réel, et ce dernier sera détaillé ci-dessous.

5. Algorithme de Viola et Jones

Une avancée majeure dans le domaine a été réalisée par les chercheurs Paul Viola et Michael Jones en 2001 [20]. Ces derniers ont proposé une méthode basée sur l'apparence (Appearance-based methods).

La méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes [20].

Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, la bibliothèque utilisée dans notre application.

Cette méthode combine quatre contributions clés qui sont les caractéristiques pseudo-haar, l'approche d'image intégrale, la méthode de Boosting et l'algorithme en cascades de classifieurs. Dans la suite, Nous allons détailler ces étapes clés tout en précisant l'apport engendré de chacun d'entre elles sur la performance et l'efficacité de la méthode.

5.1. Caractéristiques pseudo-haar

Une caractéristique pseudo-haar est représentée par un rectangle défini par son sommet, sa hauteur, sa longueur et son poids. Le recourt au traitement d'images par ces caractéristiques est motivé par sa rapidité contre le traitement direct à travers le balayage de la totalité d'image pixel par pixel.

La figure III.5 ci-dessous présente des exemples de caractéristiques utilisées par Viola et Jones. Elles sont inspirées des descripteurs de haar qui ont été employées par Papageorgiou et Al dans [21]. Les valeurs de ces caractéristiques sont calculées par la soustraction de la somme des pixels noirs de la somme des pixels blancs [22].

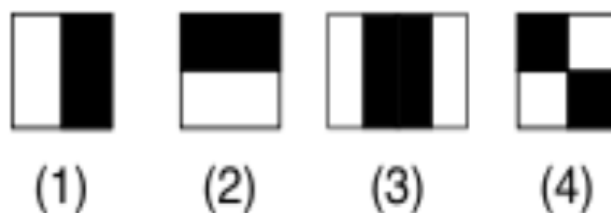


Figure III. 5: Exemple de caractéristiques pseudo-Haar [30].

L'image précédente présente quatre caractéristiques de pseudo-Haar mais il existe d'autres, et avec différentes orientations.

Pour optimiser le calcul de ces caractéristiques, une technique a été mise en place permettant de faire les opérations d'une façon extrêmement rapide. Au lieu d'utiliser l'image d'origine pour appliquer les caractéristiques Pseudo-Haar, une représentation de l'image d'origine sera générée. Cette représentation est appelée "Image Intégrale".

5.2. Image Intégrale

La méthode d'image intégrale est utilisée pour déterminer la présence ou l'absence des caractéristiques dans chaque position de l'image et à n'importe quelle taille. Le but de cette méthode à part la détection de caractéristiques est la réduction du temps de calcul de ces dernières. La valeur de l'image intégrale aux coordonnées (x, y) est la somme de tous les pixels au-dessus et à gauche de (x, y) , y compris ce dernier point (Figure III.6).

Soit ii l'image intégrale de l'image initiale i et $i(x, y)$ est la valeur de l'image intégrale au point (x, y) : [23]

On peut également définir l'image intégrale ii par:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Où $ii(x, y)$ est l'image intégrale et $i(x, y)$ est l'image originale.

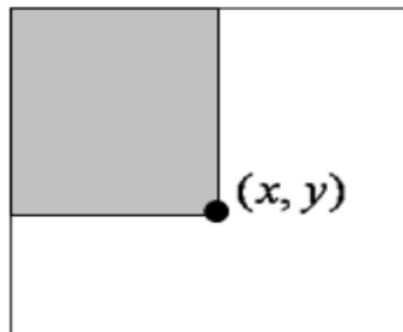


Figure III. 6: La valeur de l'image intégrale au point (x, y) [23].

5.3. Sélection de caractéristiques par boosting

Le troisième élément clé de la méthode de Viola et Jones est l'utilisation d'une méthode de boosting afin de sélectionner les meilleures caractéristiques. Le boosting est un principe qui consiste à construire un classifieur « fort » à partir d'une combinaison pondérée de classifieurs « faibles », c'est-à-dire donnant en moyenne une réponse meilleure qu'un tirage aléatoire. Viola et Jones adaptent ce principe en assimilant une caractéristique à un classifieur faible, en construisant un classifieur faible qui n'utilise qu'une seule caractéristique. L'apprentissage du classifieur faible consiste alors à trouver la valeur seuil de la caractéristique qui permet de mieux séparer les exemples positifs des exemples négatifs. Le classifieur se réduit alors à un couple (caractéristique, seuil).

L'algorithme de boosting utilisé est en pratique une version modifiée d'AdaBoost, qui est utilisée à la fois pour la sélection et pour l'apprentissage d'un classifieur « fort ». Les classifieurs faibles utilisés sont souvent des arbres de décision. Un cas remarquable, fréquemment rencontré, est celui de l'arbre de profondeur 1, qui réduit l'opération de classification à un simple seuillage [29].

L'algorithme est de type itératif, à nombre d'itérations déterminé. À chaque itération, l'algorithme sélectionne une caractéristique, qui sera ajoutée à la liste des caractéristiques sélectionnées aux itérations précédentes, et le tout va contribuer à la construction du classifieur fort final. Cette sélection se fait en entraînant un classifieur faible pour toutes les caractéristiques et en sélectionnant celui avec l'erreur la plus faible sur l'ensemble d'apprentissage. L'algorithme tient également à jour une distribution de probabilité sur l'ensemble d'apprentissage, réévaluée à chaque itération en fonction des résultats de classification. En particulier, plus de poids est attribué aux exemples difficiles à classer, c'est-à-dire ceux dont l'erreur est élevée. Le classifieur « fort » final construit par AdaBoost est composé de la somme pondérée du classifieur sélectionné.

Plus formellement, on considère un ensemble de n images (x_1, \dots, x_n) et leurs étiquettes associées (y_1, \dots, y_n), qui sont telles que $y_i=0$ si l'image x_i est un exemple négatif et $y_i=1$ si x_i est un exemple de l'objet à détecter. L'algorithme de boosting est constitué d'un nombre T d'itérations, et pour chaque itération t et chaque caractéristique j , on construit un classifieur faible h_j . Idéalement, le but est d'obtenir un classifieur h qui prédise exactement les étiquettes pour chaque échantillon, c'est-à-dire $y_i = h(x_i) \forall i \in \{1 \dots n\}$ [24].

En pratique, le classifieur n'est pas parfait et l'erreur engendrée par ce classifieur est donnée par :

$$\epsilon_j = \sum_{i=1}^n w_i |h_j(x_i) - y_i|$$

Les W_i étant les poids associés à chaque exemple et mis à jour à chaque itération en fonction de l'erreur obtenue à l'itération précédente. On sélectionne alors à l'itération t le classifieur h_t présentant l'erreur la plus faible :

$$\epsilon_t = \min_j(\epsilon_j)$$

Le classifieur fort final $h(x)$ est construit par seuillage de la somme pondérée du classifieur faible sélectionné :

$$h(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{sinon} \end{cases}$$

Les α_t sont des coefficients calculés à partir de l'erreur ϵ_t .

5.4. 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. 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. 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 (c.-à-d. 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. Cette structure en cascade peut également s'interpréter comme un arbre de décision dégénéré, puisque chaque nœud ne comporte qu'une seule branche [26].

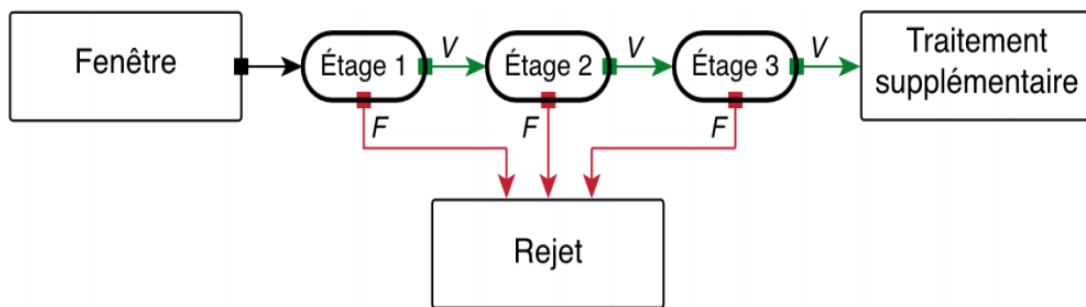


Figure III. 7: Illustration de l'architecture de la cascade : les fenêtres sont traitées séquentiellement par les classifieurs, et rejetées immédiatement si la réponse est négative (F) [26].

Une des limitations de la méthode de Viola et Jones est son manque de robustesse à la rotation, et sa difficulté à apprendre plusieurs vues d'un même objet. En particulier, il est difficile d'obtenir un classifieur capable de détecter à la fois des visages de face et de profil. Viola et Jones ont proposé une amélioration qui permet de corriger ce défaut,[25] qui consiste à apprendre une cascade dédiée à chaque orientation ou vue, et à utiliser lors de la détection un arbre de décision pour sélectionner la bonne cascade à appliquer.

Plusieurs autres améliorations ont été proposées par la suite pour apporter une solution à ce problème. [27]

5.5. Etapes clés

5.5.1. Apprentissage

L'apprentissage consiste à analyser un très grand nombre d'images positives et négatives. Plusieurs milliers d'exemples sont nécessaires, et leur analyse prend plusieurs semaines. L'apprentissage comprend :

- Le calcul des caractéristiques pseudo-Haar (positifs et négatifs)
- L'entraînement de la cascade : pour chaque étage, on crée un classifieur fort, par ajout successif de classifieurs faibles entraînés sur la même caractéristique. On ajoute ces classifieurs faibles jusqu'à obtenir un classifieur fort possédant la performance souhaitée (taux de détection/faux positifs) pour l'étage [51].

5.5.2. Détection

La détection est l'étape qui s'applique sur l'image dans laquelle on souhaite déceler la présence d'objet.

On parcourt l'ensemble de l'image à toutes les positions et échelles, avec une sous fenêtre de 24x24 pixels. On applique la cascade à chaque sous fenêtre :

- Calcul des caractéristiques pseudo-Haar
- Calcul de la réponse du classifieur.
- Passage à l'étage supérieur si la réponse est positive (sous fenêtre suivante sinon).
- Si on a traversé tous les étages successivement, l'exemple est déclaré positif

Enfin, on fusionne les détections multiples qui se chevauchent [27].

5.6. Performance

Viola et Jones ont testé leur méthode sur la base de 130 images contenant 507 visage de face. Ils présentent leur résultat sous la forme d'une courbe Receiver Operating Characteristic [28]. Cette courbe présente le taux de détection correct en fonction du nombre de faux positifs. On remarque par exemple que pour un taux de détection de 88.8%, on a eu 50 faux positifs. Comparé à d'autres méthodes de détection d'objet, celle de Viola et Jones est équivalente en terme de précision de détection [27].

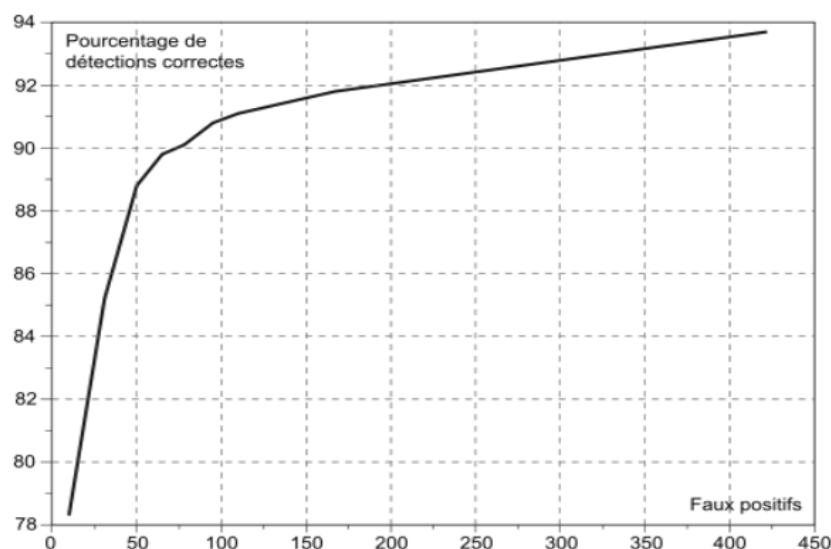


Figure III. 8: Courbe du taux de détection correct en fonction du nombre de faux positifs[27].

5.7. Applications

La méthode de Viola et Jones est principalement utilisés dans la détection de visage et la détection de personnes. Les systèmes de vidéosurveillance, d'IHM multimodales. Dans le grand pu, les appareils photos numériques utilisent cette méthode pour détecter les visages sur une photographie, et effectuer un autofocus sur les visages.

La méthode a également été utilisés pour la détection des mains dans la gestion de commandes IHM, la détection de voiture par satellite, ou pour l'évaluation du trafic sur autoroute.

6. Conclusion

La détection du visage dans une image ou dans une vidéo est un domaine de recherche très importante et aussi très difficile. Jusqu'à maintenant, des approches pour résoudre ce problème de détection de visage ont été développées en se basant sur les caractéristiques utilisé par VIOLA et JONES.

Chapitre IV

Implémentation et

Résultats

1. Introduction

Notre sujet de projet de fin d'étude, nous a permis de développement d'une application de détection des visages et des yeux dans une images ou dans une scène vidéo capturée par une webcam en utilisant la bibliothèque visuelle OpenCV et le logiciel Visual studio.

L'objectif principal de notre travail est décrit comme suite :

- Cacher automatiquement les yeux qui ont été détecté par une image.
- Brouiller automatiquement les visages qui ont été détecté par une vidéo.

2. L'objectif de travail

Le brouillage du visage vise à protéger l'identité de la personne ou la sécurité juridique de l'éditeur. Même si nous avons effectué par l'intermédiaire d'un bandeau gris sur les yeux, il apparait clairement que cette protection sommaire n'empêchait pas les proches de reconnaître le sujet photographié. Il s'agit donc d'un artifice visant à minimiser le risque juridique.

Largement pratiqué par les reportages télévisés, le brouillage intégral du visage qui s'est imposé avec l'emploi des technologies numériques semble offrir une meilleure protection de l'identité.

La pratique du brouillage est un usage typique de la presse à scandale, qui a pour but de préserver la publication d'une image, alors que celle-ci expose à un risque juridique ou révèle l'identité d'une personne connue dans les médias de manière non souhaitée.

Notre projet traite les deux cas, soit l'escompte des yeux ou le brouillage du visage.

3. Environnement du travail

Dans cette section, nous présentons les environnements matériels et logiciels de notre travail.

3.1. Environnement matériel ou le hardware

Un ordinateur Lenovo avec les caractéristiques suivantes:

- Processeur: Intel (R) Core (TM) i5-3210M CPU @ 2.50 Ghz.
- RAM: 4.00 GO.
- Disque Dur: 500 GO.
- Webcam intégrée de marque Lenovo EasyCamera.

3.2. Environnement immatériel ou le software

- Système d'exploitation : Microsoft Windows 7 - 64 bits
- Logiciel : Microsoft Visual Studio version 2008
- La bibliothèque : OPENCV version 2.3.1

4. Implémentation

4.1. Les principes de la détection du visage

4.1.1. Cascade (Classifieur)

Pour détecter les visages avec la bibliothèque OpenCV, nous utilisons le fichier XML qui contient les principales caractéristiques d'un visage plus précisément le détecteur HaarCascade (connu comme la méthode Viola-Jones).

OpenCV est livré avec plusieurs classifieurs différents pour la détection du visage, la détection des yeux, la détection de bouche, la détection du corps....etc. Ces différents classifieurs sont situés dans le dossier « data/haarcascades » qui se trouve dans le répertoire OpenCV.

Le classifieur qui est utilisé pour la détection du visage c'est

« haarcascade_frontalface-alt.xml » Voici le code source

```
string face_cascade_name = "c:/haarcascade_frontalface_alt.xml";  
CascadeClassifier face_cascade;
```

```
string window_name = "Capture - Face detection";  
int filenumber; // Number of file to be saved  
string filename;
```

4.1.2. Les opérations pour lire les images ou les vidéos

➤ *Pour les images*

Voici le code source qui est utilisé pour charger une image stockée dans l'ordinateur

```
// Read the image file  
Mat frame = imread("image1.jpg");  
for ( ; ; )  
{  
    // Apply the classifier to the frame  
    if (!frame.empty())  
{  
        detectAndDisplay(frame);  
    }  
    else  
    {  
        printf(" --(!) No captured frame -- Break!");  
        break;  
    }  
  
    int c = waitKey(10);  
    if (27 == char(c))  
    {  
        break;  
    }  
}
```

➤ *Pour les vidéos*

En ce qui suit, le code source utilisé pour la lecture de la vidéo

```
// Read the video stream
Mat frame;
for (;;)
{
capture >> frame;
// Apply the classifier to the frame
if (!frame.empty())
{
detectAndDisplay(frame);
}
else
{
printf(" --(!) No captured frame -- Break!");
break;
}
int c = waitKey(10);
if (27 == char(c))
{
break;
}
}
```

4.1.3. La fonction de traitement et de détection

DetectAndDisplay () elle correspond à la fonction principale du programme implémenté, permet d'effectuer tous les traitements nécessaires.

DetectMultiScale () cette fonction est utilisée dans le cœur de la fonction précédente pour le principe de détection du visage, ci-dessous le code source.

```
face_cascade.detectMultiScale(frame_gray, faces, 1.1, 2, CASCADE_SCALE_IMAGE,
Size(30, 30));
```

4.1.4. Le traçage de rectangle autour du visage

La structure de cette fonction est définie dans le programme implémenté et elle ne s'exécute qu'après avoir déterminé ces paramètres (pt1, pt2) qui se déduisent à partir du traitement effectué sur les images.

```
// Display detected faces on main window
```

```
Point pt1(faces[ic].x, faces[ic].y);
Point pt2((faces[ic].x + faces[ic].height), (faces[ic].y + faces[ic].width));
rectangle(frame, pt1, pt2, Scalar(0,255, 0), 4, 8, 0);
```

4.1.5. La fenêtre du résultat

Cette fenêtre est destinée à afficher le résultat de détection de l'algorithme utilisé. Parmi les actions faites sur cette fenêtre nous trouvons

- *Imshow (nom fenêtre, image)* : affichage dans l'interface du résultat.
- *cvDestroyWindow (nom de la fenêtre)* : permet de détruire la fenêtre de capture, elle permet aussi de libérer la capture.

Le code source utilisé :

```
// Show image
```

```
sstm << "Crop area size: " << roi_b.width << "x" << roi_b.height << " Filename: " <<
filename;
text = sstm.str();
putText(frame, text, cvPoint(30, 30), FONT_HERSHEY_COMPLEX_SMALL, 0.8,
cvScalar(0, 0, 255), 1, CV_AA);
imshow("original", frame);
if (!crop.empty())
{
    imshow("detected", crop);
}
else
destroyWindow("detected");
```

4.2. Test et résultat de la détection du visage

Les deux figures ci-dessous permettent d'afficher les résultats de la détection du visage. La première (Figure IV.1) pour la détection du visage dans une image stockée sur l'ordinateur, et la deuxième (Figure IV.2) pour la détection du visage dans une vidéo capturée par une webcam.

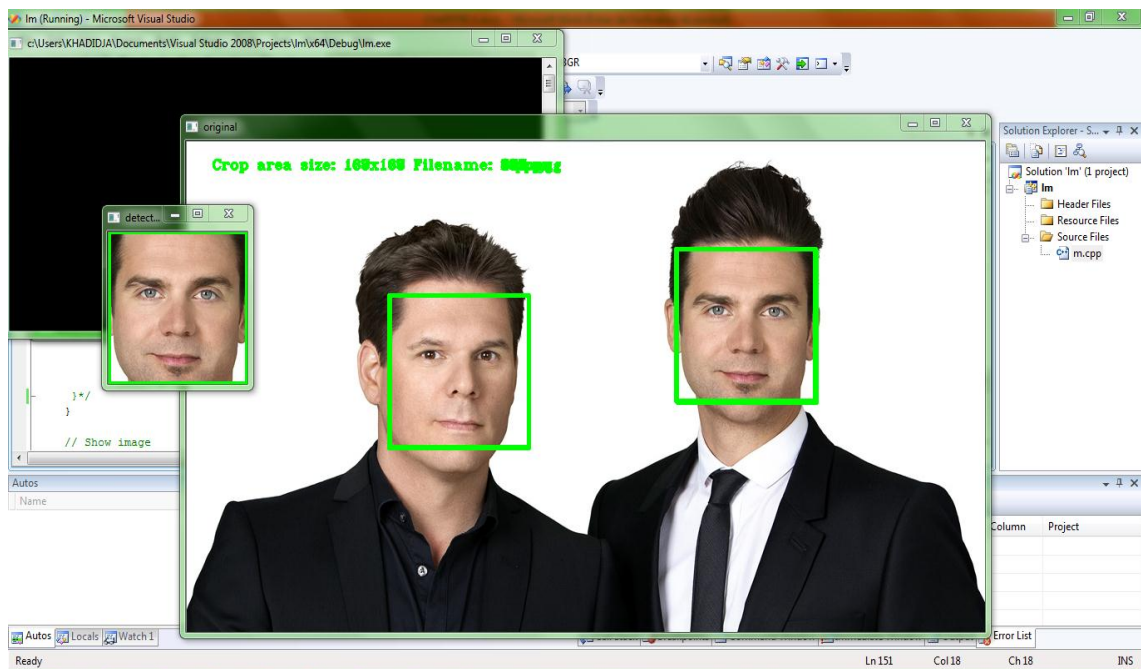


Figure IV. 1 : détection du visage dans une image.

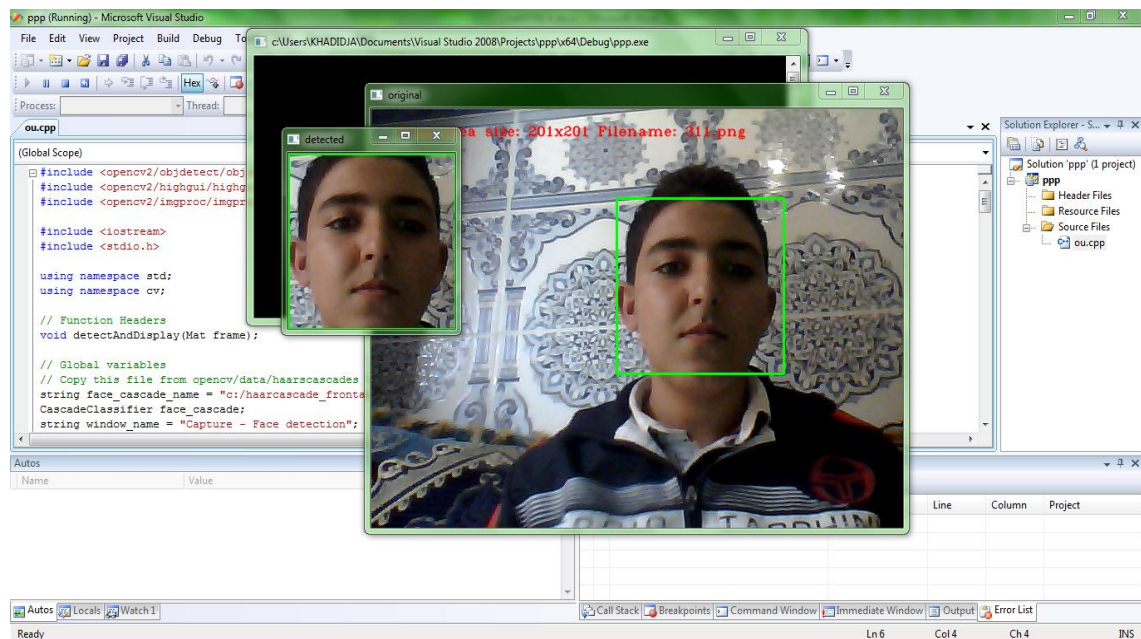


Figure IV. 2 : détection du visage dans une vidéo.

4.3. Les principes de la détection des yeux

4.3.1. Cascade (Classifieur)

Le classifieur utilisé pour la détection des yeux est

« haarcascade_eye_tree_eyeglasses.xml »

Le code source:

```
String eyes_cascade_name = "c:/haarcascade_eye_tree_eyeglasses.xml";
CascadeClassifier eyes_cascade;
```

4.3.2. La fonction de détection des yeux

eyes_cascade.detectMultiScale(): cette fonction est utilisée au coeur de la fonction du traitement *DetectAndDisplay ()*.

4.3.3. Le traçage de rectangle autour de l'œil

Utilise les mêmes propriétés de la fonction de traçage du visage.

Ci-dessous le code source pour la détection des yeux :

```
Mat faceROI = frame_gray( faces[ic] );
    std::vector<Rect> eyes;

    //-- In each face, detect eyes
    eyes_cascade.detectMultiScale( faceROI, eyes, 1.1, 2, 0 |CV_HAAR_SCALE_IMAGE,
    Size(30, 30) );
    for( int j = 0; j < eyes.size(); j++ )
    {
    Point pt1(faces[ic].x + eyes[j].x + eyes[j].width, faces[ic].y + eyes[j].y + eyes[j].height);
    Point pt2(faces[ic].x + eyes[j].x, faces[ic].y + eyes[j].y);
    rectangle(frame, pt1, pt2, Scalar(0, 210, 255, 0), 4, 8 ,0);
    }
}
```

4.4. Test et résultat de la détection des yeux

Voici le résultat obtenu pour la détection des yeux dans une image (Figure IV.3) et dans une vidéo (Figure IV.4).

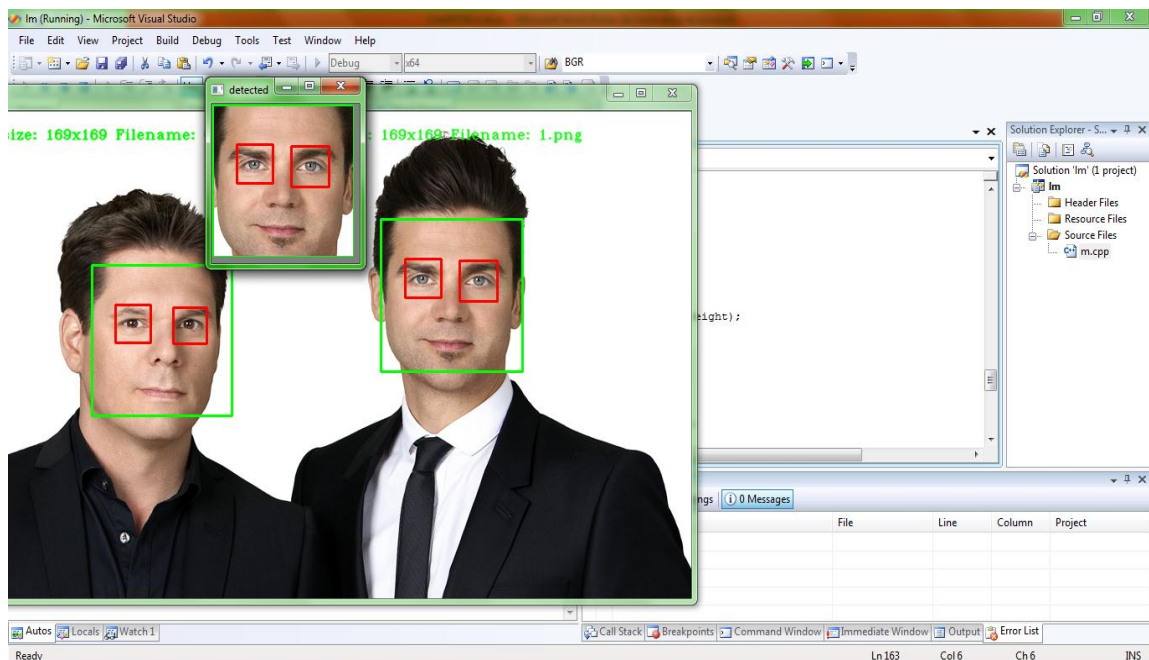


Figure IV. 3 : détection des yeux dans une image.

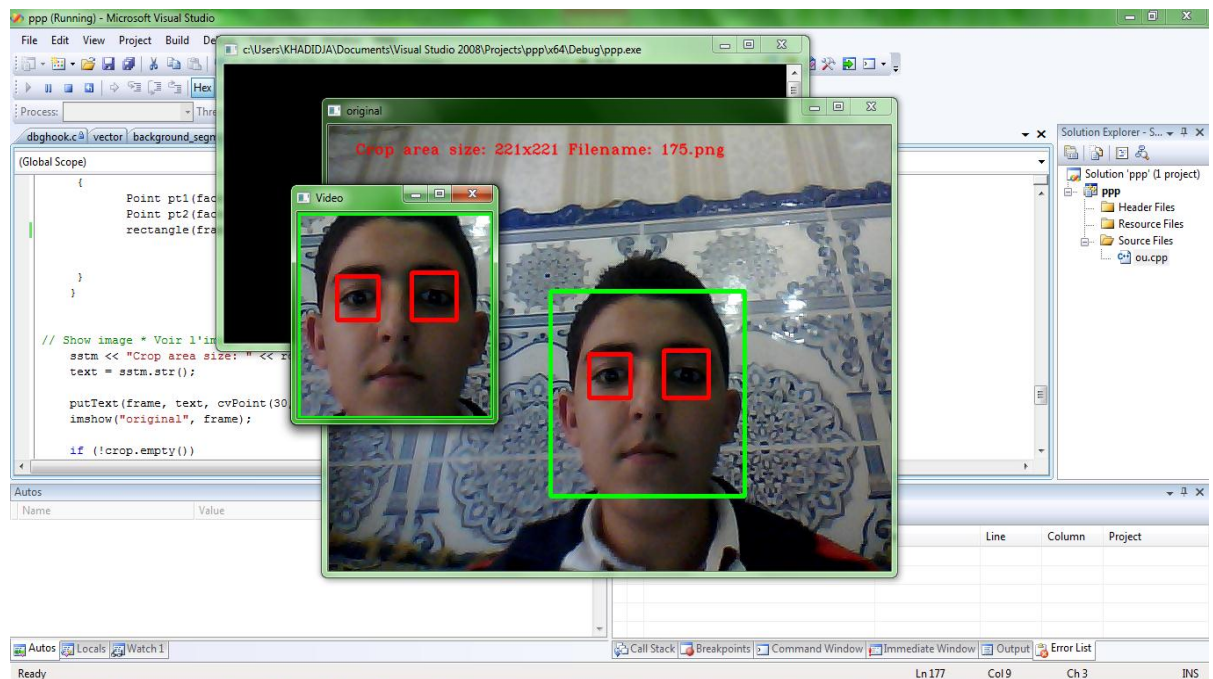


Figure IV. 4: détection des yeux dans une Vidéo.

4.5. Cacher les yeux

Pour masquer les yeux dans une image numérique nous changeons dans les paramètres de la fonction de traçage de rectangle autour de l'œil. Voici le code source qui est modifié.

```
//-- In each face, detect eyes
```

```
eyes_cascade.detectMultiScale( faceROI, eyes, 1.1, 2, 0 |CV_HAAR_SCALE_IMAGE,
Size(30, 30) );
```

```
for( int j = 0; j < eyes.size(); j++ )
```

```
{
```

```
    Point pt1(faces[ic].x + eyes[j].x + eyes[j].width, faces[ic].y + eyes[j].y + eyes[j].height);
```

```
    Point pt2(faces[ic].x + eyes[j].x, faces[ic].y + eyes[j].y);
```

```
        rectangle(frame, pt2, pt1, Scalar(175, 175,175), 45, 8, 0);
```

```
}
```

L'affichage du résultat obtenu dans la figure IV.5

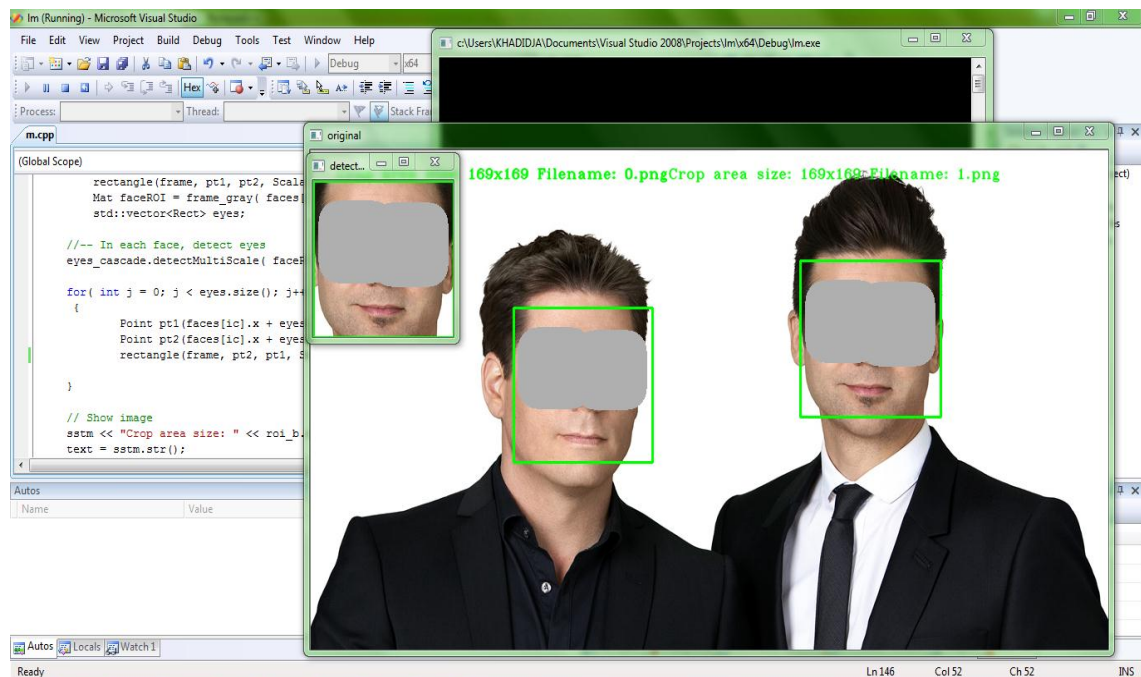


Figure IV. 5: Cacher les yeux.

4.6. Brouillage du visage

Pour brouiller le visage détecté dans une vidéo capturée par une webcam nous utilisons les techniques de la conversion d'un domaine de couleur original 'RGB' à l'espace de couleur HSV expliqué dans le troisième chapitre.

Le code source pour flouter le visage et le résultat obtenu dans notre application :

// Show image * Voir l'image

```
sstm << "Crop area size: " << roi_b.width << "x" << roi_b.height << " Filename: " << filename;
text = sstm.str();
putText(frame, text, cvPoint(30, 30), FONT_HERSHEY_COMPLEX_SMALL, 0.8,
cvScalar(0, 0, 255), 1, CV_AA);
imshow("original", frame);
if (!crop.empty())
{
    cvtColor(crop, crop, CV_BGR2HSV);
    resize(frame, frame, Size(), 1, 1);
    imshow("Video Brouiller", frame);
```

```

        waitKey(5);
    }
else
    destroyWindow("detected");
}

```

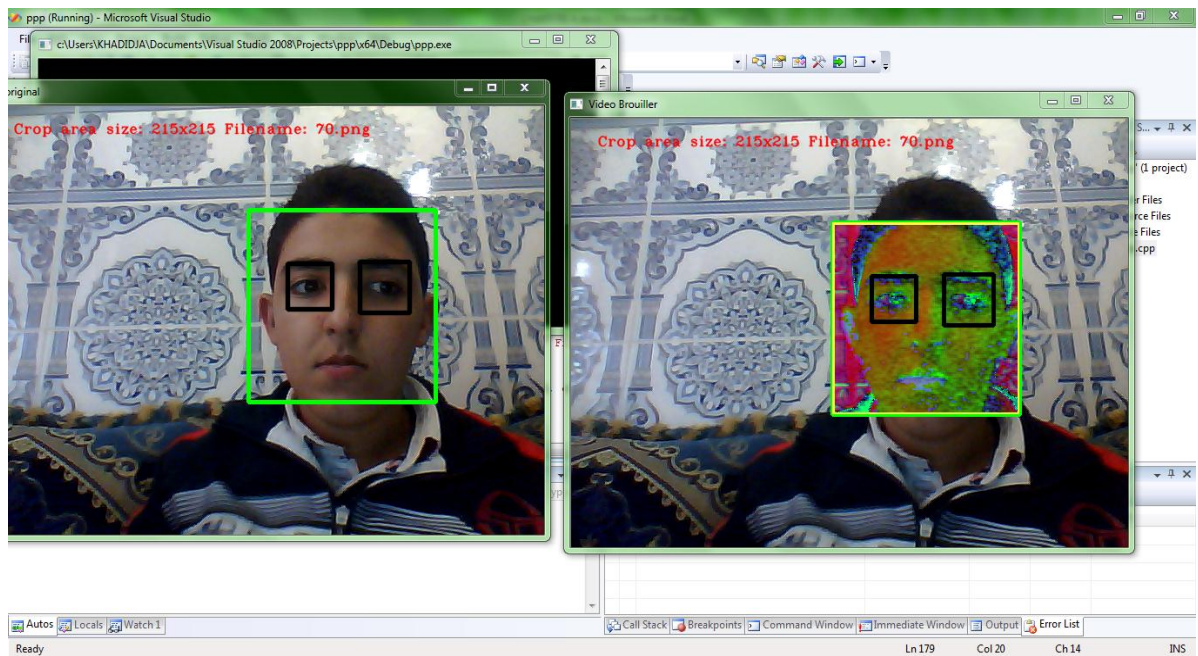


Figure IV. 6: Brouillage du visage

5. Conclusion

Nous avons réussi à réaliser un système de deux applications,

La première se résume dans la détection du visage dans une vidéo capturée par la webcam de l'ordinateur pour le brouiller.

La deuxième consiste à la détection du visage et des yeux dans une photo pour escompter les yeux.

Nous avons choisi pour chaque application des méthodes adaptées à notre système, et les résultats obtenus ne sont pas fiables à 100% mais nous avons atteint l'objectif de notre projet.

Conclusion Générale

Conclusion générale

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de réaliser une application de détection du visage et des yeux puis brouiller le visage et cacher les yeux.

Au cours de cet mémoire, nous avons étudié et implémenté l'algorithme « Viola et Jones » pour la détection du visage et des yeux et nous avons intégré les techniques de l'espace de couleur HSV dans le brouillage du visage.

L'algorithme de « Viola et Jones » semble être la meilleure application en utilisant l'architecture de cascade et l'image intégrale. Profitant des avantages de cette nouvelle technique qui rapporte une performance primordiale dans la détection du visage en temps réel.

La bibliothèque OpenCV est un élément important dans la réalisation et le développement de notre projet et cela revient à la richesse de différentes fonctions qui le compose surtout dans le traitement des images et des vidéos.

Comme perspective de notre travaille, nous souhaitons pour les prochains projets de fin d'étude, élargir notre travail de détection du visage pour faire une comparaison entre algorithme Viola et Jones qui a été implémenté avec d'autres méthodes qui existent dans ce domaine.

Bibliographie

Références

- [1]: P.Nohar, Introduction à la vision par ordinateur, 28/03/2012.
- [2]: R.C.Gonzalez, P.Wintz, Digital Image Processing, ADDISON-WESLEY, second edition, 1987.
- [3]: J.P.GUILLOIS, techniques de compression des images.
- [4]: M^f Cherifi Mehdi, Application de la transformation de radon discrète pour la compression d'images MSG, Mémoire de magister en électronique, Université Mouloud Mammeri, Tizi-Ozou, Faculté de génie électrique et de l'informatique, 05/03/2015.
- [5]: Alain et Ursula Bouteville, “ La télévision numérique ”, 4^{ème} Edition Dunod, Paris, 2006.
- [6]: D. A. Huffman, “A method for the construction of minimum - redundancy codes”, Pro -ceedings of the Institute of Radio Engineers, vol. 40, pages 1098–1101. Sept. 1952.
- [7]: Kamel Moucits, Les Fichier Vidéo une problématique en Milieu Hétérogène, Université Henri Poincaré-Nancy, 23/01/2011.
- [8]:Alain Filhol, Introduction au monde merveilleux de la video numérique, Institut Laue-Langevin, 9 nev 2009.
- [9]: M^f Slimane Larbi, Systèmes Multimédia, Université des sciences et de la technologie Houari Boumediene, 2014.
- [10]: Fabrice Urban. Implan tation optimisée d'estimateurs de mouvement pour la compression vidéo sur plates-formes hétérogènes multi composants. Signal and Image processing. INSA de Rennes, 2007. French.
- [11]: Xin Zou. Illumination Invariant Face Recognition Based in Active Near-Infrared Die rental Imaging. PhD thesis, University of surrey, March 2007.

Bibliographie

- [12]: G. C. Feng and P. C. Yuen. Variance projection function and its application to eye detection for human face recognition. *Pattern Recogn.Lett.*, 19(9), 1998.
- [13]: Saad A. Sirohey and Azriel Rosenfeld. Eye detection in a face image using linear and nonlinear lters. *Pattern Recognition*, 34:13671391, 2001.
- [14]: Walid Hizem, capteur intelligent pour la reconnaissance de visage, thèse présentée pour l'obtention du diplôme de docteur, l'université pierre et Maria Curie Paris 6, 2009.
- [15]: OUAFEK REDOUANE, MEGHALSI MOHAMED DEDA, la détection des visages dans une image, 2008/2009.
- [16]: Zrelli Mohamed, Implémentation d'une méthode de détection et suivi de visage en temps reel, Travail de fin d'étude, Ecole Royale Militaire Belgique.
- [17]: BENCHERT, BOUALLEG, TEBIKH: Segmentation de la couleur de peau par seuillage selon différents espaces couleurs, 3ème journée internationale sur l'informatique graphique en TUNISIE.
- [18]: Vladimir Veznevets, "A survey on pixel Based skin color detection techniques", Ph.D.Teses, Faculty of computational mathematics and cybernetics Moscow state university, Moscow, 2005.
- [19]: Noureddine BESBES, INDEXATION EN INTERVENANT D'UN DOCUMENT VIDÉOPAR IDENTIFICATION DU VISAGE, MÉMOIRE du Master Recherche (Image, Information, Hypermédia), UNIVERSITE PAUL SABATIER TOULOUSEIII, 26/06/2007.
- [20]: Paul Viola and Michael Jones. Robust real-time object detection. In *Second international work shop on statistical and computation atheories of vision*, Van-couver, Canada, July 13 2001.
- [21]: C. Papageorgiou, M. Oren ET T. Poggio, "A *General Framework for Object Detection*", *International Conference on Computer Vision*, pages 555-562, Janvier 1998.

Bibliographie

- [22]: P.A. Negri, “*Détection et Reconnaissance d’objets structurés: Application aux Transports Intelligents*”, Thèse, Université Pierre et Marie Curie - Paris VI, France, Septembre 2008.
- [23]: M^r Oudjani Brahim, detection du visage par la method Hybride Ada Boost et couleur de la peau, Mémoire de Magister, Université de Gulma faculté de science et de l’ingénierie.
- [24]: Jones et P. Viola, « *Fast Multi-view Face Detection* », IEEE CVPR, 2003.
- [25]: Paul Viola et Michael Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, IEEE CVPR, 2001.
- [26]: A. L. C. Barzak, To Ward and Efficient Implementation of a Rotation Invariant Detection using Haar-like Features. Proceeding of the 2009 IEEE/RSJ international conference on Intelligent robotsand systems. Nouvelle-Zélande, Dunedin.2005, P31-36.
- [27]:H.Rowley, S.Baluja et T.Kanade, « Neural network-based face detection », IEEE Transactions on Patten Analysis and Machine Intelligence, 1998.
- [28]: <http://fr.openclassrooms.com/informatique/cours>
- [29]:http://www.memoireonline.com/01/14/8585/m_Identification-des-personnes-par-reconnaissance-de-visage-pour-la-securite-d-une-institution-banca19.html
- [30]:<http://fr.wikipedia.org/wiki/>
- [31]: www.kaddour.com/chap1/chap1.html
- [32] :www.mysic-assoc.net/spip/IMG/pdf/te-traitement-d-image-2.pdf
- [33]:<http://www.shootingdesign.com/articles/article4/understanding-resolution-photoshop.html>
- [34]: <http://bu.umc.edu.dz/theses/informatique/TOR4276.pdf>
- [35]: <http://djoume.taket.org/infoc/TP5.pdf>

Bibliographie

- [36] http://www.unit.eu/cours/videocommunication/Transformation_ponctuelle_histogramme.pdf
- [37]: www.lagis.univ-lille1.fr/formations/sii/06-07/projets/cmucam-rapport.pdf
- [38]: <http://www.unise.fr/perso/favier/vision/traitements-ponctuels.pdf>
- [39] <http://images.math.cnrs.fr/Le-traitement-numerique-des-images.html>
- [40]: <http://www.journaldunet.com/developpeur/tutoriel/gra/>
- [41]: <http://commentcamarche.net/>
- [42]: <http://www.europschool.net/static/formation/>
- [43]: http://duson.perso.sfr.fr/Telechargt/Formats_fichiers_Video.pdf
- [44]: www.axis.com/fr/
- [45]: http://pod.univ-lille1.fr/media/filer_public/8b/ee/8bee78fc-4123-4078-8579e6ea50462eac/huin_synthese.pdf
- [46]: http://www.sen-av.net/article.php3?id_article=28
- [47]: http://fr.wikipedia.org/wiki/D%C3%A9tection_de_visage
- [48]: <http://www.memoireonline.com/>
- [49]: <https://www.uclouvain.be/cps/ucl/doc/commission-gbio/documents/Thesis2.pdf>
- [50]: <https://www.youtube.com/watch?v=bSeFrPrqZ2A>
- [51]: <http://pierreguillot.fr/wordpress/wpcontent/uploads/2013/03/TPreconnaissancefaciale.pdf>
- [52]: [http://www.gypsevideo.fr/index.php?option=com_content&view=article&id=123:exportation&catid=51:lire-la-suite.](http://www.gypsevideo.fr/index.php?option=com_content&view=article&id=123:exportation&catid=51:lire-la-suite)
- [53]: <http://www.hd-motion.com/>
- [54]: http://people.rennes.inria.fr/Aline.Roumy/teaching/14-Compression_images_video.pdf
- [55]: <http://fr.slideshare.net/MedAymenFodda/reconnaissance-faciale-45358788>