Table des matières

INTRODUCTION GENERALE 10

Chapitre1 : Reconnaissance Des Formes

1. INTRODUCTION 13
2. LA RECONNAISSANCE DES FORMES 13
2.1 DEFINITION 13
2.2 HISTORIQUE 14
3. METHODES 15
3.1 METHODES DE RECONNAISSANCE DE FORMES : 15
4. LA RECONNAISSANCE DE PLUSIEURS OBJETS DANS UNE IMAGE 16
5. APPLICATIONS TYPIQUES DE LA RECONNAISSANCE DES FORMES 16
6. SCHÉMA GÉNÉRAL D'UN SYSTÈME DE RECONNAISSANCE DES FORMES 17
6.1 PRÉPARATION DES DONNÉES 17
a. Numérisation. 17
b. Prétraitement. 17
c. Calcul des représentations 18
6.2 Apprentissage 18
6.3 Classification 18
6.4 Post traitement. 18
7. TRAITEMENT D'IMAGE 18
7.1 PRÉTRAITEMENT OU TRAITEMENT BAS NIVEAU 20
7.1.1 DÉFINITION D'IMAGE 20
7.1.2 DÉFINITION PIXEL 21
7.1.3 IMAGE EN NIVEAU DE GRIS 21
-FILTRE GAUSSIEN 22
-DU CONTINU AU DISCRET 22
8. SEGMENTATION DES IMAGES 24
8.1 LES PRINCIPES DE LA SEGMENTATION 24
a. La segmentation par régions 25
b. La segmentation par seuillage 26
b.1 Détection automatique du seuil 27
b.2 Seuillage à hystérésis 27
c. La segmentation par contours 28
c.1 Détection de contours 28
c.2 Détection de contours par la méthode de Canny 29
d. La transformée de Hough 30

9. CONCLUSION 34

Chapitre 2: *Mise en place d'une application de Reconnaissance de Formes*

1. INTRODUCTION 38
2. PRESENTATION DE LA LIBRAIRIE OPENCV 38
2.1 HISTORIQUE 38
2.2 QU'EST-CE QUE OPENCV 39
2.3 FONCTIONNALITES 39
- Traitement d'images 39
- Traitement vidéo 39
2.4 MODULES OPENCV 40
- CxCore (Coeur d'OpenCV) : 40
- HighGui (librairie d'interfaçage graphique) 40
- ImgProc (traitement d'image) : 40
- Vidéo (traitement de flux vidéo) 40
- ObjDetect (détection d'objets) 40
- ML (librairie d'apprentissage automatique) 40
- Calib3D 40
2.5 MICROSOFT VISUAL STUDIO 2010 41
3 ENVIRONNEMENT DU TRAVAIL 41
3.1 ENVIRONNEMENT MATERIEL OU LE HARDWARE: 41
3.2 ENVIRONNEMENT IMMATERIEL OU LE SOFTWARE : 41
4 DIAGRAMME DE CLASSES 42
5 INTERFACES DE L'APPLICATION 43
6 CONCLUSION 47
CONCLUSION GENERALE 48
References bibliographiques 50

Table des figures

Figure 1.1: La méthode de reconnaissance de multi objets dans une image 16

Figure 1.2: Schéma général d'un système de reconnaissance des formes 17

Figure 1.3: Données quantitatives des images 19

Figure 1.4 : Transformation image RGB en image en niveau de gris. 19

Figure 1.5 : Exemple de réseau de pixels 20

Figure 1.6: Filtre de bruit gaussien 23

Figure 1.7: La méthodeOTSU seuil=136 Figure 8.1.8: La ethod OTSU seuil=102 27

Figure 1.9: Seuillage par hystérésis 28

Figure 1.10: Différentes formes de contours 29

Figure 2.1 : Diagramme de Classe en UML 42

Figure 2.2 : Interface d'accueil de l'application Reconnaissance de Formes 43

Figure 2.3 : L'interface de notre application par la selection Fichier d'image. 44

Figure 2.4 : L'interface de notre application par la selection WebCam. 45

Figure 2.5 : Détection de la forme d'objets en mouvement. 46

INTRODUCTION

INTRODUCTION GENERALE

La reconnaissance des formes a pour objet de simuler les activités de perception sensorielle du cerveau. En premier lieu, la perception visuelle et auditive, y compris dans des bandes spectrales non perçues par l'homme (infra rouge, radar, sonar, ...). La reconnaissance a besoin d'un modèle de l'objet. Pour l'être humain, ce modèle correspond à une représentation mentale de l'objet qui peut être apprise en retenant les caractéristiques les plus discriminantes de l'objet. Les caractéristiques peuvent être toutes sortes d'attributs de l'objet : forme, couleur, texture, taille, volume, etc.

Objet de recherches depuis l'apparition de l'informatique, cette discipline prend désormais un essor considérable dans l'industrie, car les progrès de l'électronique permettent désormais de doter les systèmes opérationnels de fortes puissances de calcul à des prix raisonnables. Elle regroupe un certain nombre de techniques allant de la simple identification d'objets isolés à l'analyse de scènes complexes ou la compréhension de la parole. Dans ces derniers cas, il faut tout à la fois segmenter en objets, identifier ces objets et établir un réseau de relations entre objets. La notion d'apprentissage est au cœur de la plupart des techniques développées. Ces techniques peuvent être utilisées soit seules dans le cas de systèmes dont la mission principale est d'identifier des objets, soit en relation étroite avec des techniques d'intelligence artificielle dans le cas de systèmes devant à la fois percevoir et raisonner sur les choses perçues.

En d'autres termes, le problème que cherche à résoudre la reconnaissance des formes est d'associer une étiquette à une donnée qui peut se présenter sous la forme d'une image brute ou d'un signal. Des données différentes peuvent recevoir la même étiquette ; ces données sont les réalisations ou les exemplaires de la classe identifiée par l'étiquette. Par exemple, l'écriture manuscrite du caractère A varie d'un scripteur à l'autre mais le lecteur identifiera le caractère A pour chacune de ces réalisations.

Des méthodes générales ont été développées en reconnaissance des formes pour extraire automatiquement des informations des données sensibles afin de caractériser les classes de formes (apprentissage) et d'assigner automatiquement des données à ces classes (reconnaissance). La mise en œuvre de ces méthodes générales pour des problèmes particuliers amène à introduire la notion de processus de reconnaissance qui pose la question de l'intégration des méthodes de la reconnaissance de formes dans un système qui a pour but de reconnaître des formes.

Parallèlement aux travaux sur les méthodes de reconnaissance, se développaient le traitement d'image et la vision par ordinateur. Ces domaines ont focalisé le problème de la reconnaissance sur des données spécifiques, mais par ailleurs ils ont permis de situer la reconnaissance dans un processus plus vaste d'interprétation d'image impliquant des niveaux de perception et des connaissances propres au domaine.

Dans ce traail de PFE nous nous sommes posés comme objectif de modéliser et concevoir une application de reconaissance de forme qu'on utilisera pour détecter certaines formes via une vidéo.

Pour se faire notre manuscrit est partagé en 2 chapitres, le premier traite des notions de base attenantes à notre thématique et le deuxième récapitule toutes les étapes de modélisation et conception de notre application.

Chapitre 1

RapportGratuit.com

Chapitre1 : Reconnaissance Des Formes

1. INTRODUCTION

L'un des principaux objectifs de l'ordinateur est d'automatiser les tâches habituellement effectuées par l'homme. La plupart des activités humaines reposent sur une faculté importante de notre cerveau: pour voir distinguer entre les formes. Par exemple, reconnaître une pomme d'une orange ; un bon diagnostic d'un mauvais, ou encore un sous-marin d'une baleine. Pour pouvoir automatiser ce genre de tâches, l'ordinateur doit obligatoirement acquérir la faculté de reconnaître les formes. Dans ce chapitre, nous allons passer en revue quelques techniques des plus connues permettant la reconaissance des formes pour en choisir une que nous intègrerons dans notre application

2. LA RECONNAISSANCE DES FORMES

2.1 DEFINITION

On désigne par **reconnaissance de formes** (ou parfois **reconnaissance de motifs**) un ensemble de techniques et méthodes visant à identifier des *motifs* à partir de données brutes afin de prendre une décision dépendant de la catégorie attribuée à ce motif. On considère que c'est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique et aux statistiques.

Les *formes* ou *motifs* à reconnaître peuvent être de nature très variée. Il peut s'agir de contenu visuel (code barre, visage, empreinte digitale…) ou sonore (reconnaissance de parole), d'images médicales (rayon X, EEG, IRM…) ou multi spectrales (images satellitaires) et bien d'autres.

Watanabe [1] a défini une forme comme: « *l'opposé du chaos; c'est une entité vaguement définie, à laquelle ont peut associer un nom* ». En des termes informatiques, une forme est un ensemble de valeurs, appelés *attributs*, auxquels est associé un nom (ou étiquette), qui est leur *classe***.** Plusieurs formes peuvent avoir la même classe, on dit alors que ce sont les *exemples* ou *réalisations* de la classe.

Le problème que cherche à résoudre la reconnaissance des formes est d'associer une classe à une forme inconnue (qui n'a pas encore de classe associée). On considère souvent la Reconnaissance des formes comme un problème de classification: trouver la fonction qui affecte à toute forme inconnue sa classe la plus pertinente. Elle est partie intégrante de tout système intelligent destine à la prise de decision [2].

2.2 HISTORIQUE

Or que ce soit pour déchiffrer un texte dactylographié ou manuscrit, pour compter des chromosomes, reconnaître une tumeur, un char ou un avion de guerre, la compréhension de l'image, sa classification passe toujours par la reconnaissance d'une forme. « *Plusieurs approches théoriques ont été développées* », explique *Olivier Faugeras*.

« Les premières consistaient à faire des calculs à partir de l'image et construire des représentations symboliques de plus en plus complexes, d'abord en deux dimensions tel que sur l'image, puis tridimensionnelles, pour tenter de restituer une description proche de notre propre vision. » Un peu partout dans le monde, les chercheurs ont mis au point des méthodes mathématiques permettant de détecter les contours des objets à partir des changements rapides de contraste dans l'image, des ombres et des lumières, des régions homogènes en couleur, en intensité, en texture.

« Dès 1964, des chercheurs français, Georges Matheron (1930-2000) et Jean Serra, ont développé une autre approche théorique (baptisée morphologie mathématique) et un outil spécifique (l'analyseur de texture breveté en 1965, ndlr) d'abord pour analyser des microphotographies de terrain et évaluer des teneurs en minerai, puis pour d'autres applications comme la cytologie (caractérisation et comptage de cellules) » rappelle Olivier Faugeras. En 1968, ils créent le Centre de morphologie mathématique de l'Ecole des Mines de Fontainebleau. Leurs outils d'analyse et d'interprétation d'images sont longtemps restés franco-français, jusqu'à ce qu'un américain, Robert Haralick (Université du Kansas à cette époque, de Seattle actuellement), en fasse une large publicité dans les années 1980, en les adaptant à de nombreuses applications : industrielles comme l'inspection radiographique des ailes d'avions de Boeing, aériennes ou médicales [19].

D'autres chercheurs, comme les américains Marvin Minsky et Seymour Papert du MIT (Massachussets Institute of Technology) ont considéré le problème dans l'autre sens, en cherchant à formaliser et à faire reproduire par l'ordinateur notre propre processus de reconnaissance d'images, donc notre propre vision. Cette démarche était dans l'air du temps, au cœur des promesses de « l'intelligence artificielle » qui devait permettre de mettre l'intelligence en équations et doter les ordinateurs de toutes les capacités humaines de raisonnement, mémoire, perception. Or la vision s'est révélée être un domaine particulièrement complexe à modéliser tant elle est basée sur une quantité phénoménale de connaissances à priori fondée sur notre intelligence et notre expérience [3].

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

3. METHODES

La reconnaissance de motifs peut être effectuée au moyen de divers algorithmes d'apprentissage automatique tels:

- un réseau de neurones
- une analyse statistique
- l'utilisation de modèles de Markov cachés
- une recherche d'isomorphisme de graphes ou sous-graphes

Les formes recherchées peuvent être des formes géométriques, descriptibles par une formule mathématique, telles que :

- cercle ou ellipse
- courbes de Bézier, splines
- droite

Elles peuvent aussi être de nature plus complexe:

- lettre
- chiffre
- empreinte digitale

Les algorithmes de reconnaissance peuvent travailler sur des images en noir et blanc, avec en blanc les contours des objets se trouvant dans l'image. Ces images sont le fruit d'algorithmes de détection de contours. Ils peuvent aussi travailler sur des zones de l'image prédéfinies issues de la segmentation de l'image.

3.1 METHODES DE RECONNAISSANCE DE FORMES :

- Mise en correspondance de graphes
- Méthode Bayesienne
- Estimation Paramétrique
- Classifieur linéaire
- Réseau de neurones
- Local feature focus
- SVM : Support Vector Machine
- Polytôpes de contrainte
- Méthode des hypercubes

Un algorithme bien connu pour la détection de formes, la transformée de Hough, est une méthode d'estimation paramétrique.

4. LA RECONNAISSANCE DE PLUSIEURS OBJETS DANS UNE IMAGE

Une seule image peut être constituée d'un ou plusieurs objets [4]. Dans le cas où on désire détecter plusieurs objets dans une même image, on peut utiliser le procédé de reconnaissance multi objets représenté sur la figure 1.1 ci- dessous.

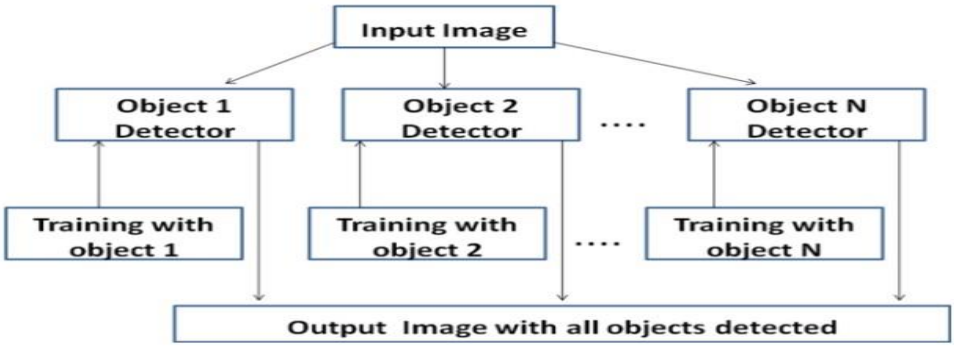


Figure 1.1: La méthode de reconnaissance de multi objets dans une image

5. APPLICATIONS TYPIQUES DE LA RECONNAISSANCE DES FORMES

- **Marketing:** La reconnaissance des formes est souvent utilisée pour classer les consommateurs selon les produits qu'ils sont susceptible d'acheter. Elle est aussi utilisée par les sociétés de vente pour classer les clients selon qu'ils soient de bons ou mauvais payeurs, ou encore selon qu'ils vont oui ou non passer à la concurrence [5].
- **Finances:** les systèmes de reconnaissance des formes sont utilisés pour la détection de transactions bancaires frauduleuses ainsi que la prédiction des banqueroutes [6].
- **Usinage:** la qualité des produits dépend souvent de paramètrisation correcte, et les relations exactes entre la qualité et les valeurs des paramètres n'est pas claire. Les systèmes de reconnaissance des formes sont utilisés pour classer les paramètres selon la qualité des produits qu'ils sont susceptibles de générer. Ils permettent ainsi de réduire le nombre d'essais ce qui fait gagner du temps et de l'argent [7].
- **Energie:** les systèmes de reconnaissance des formes sont utilisés pour prévoir la consommation électrique (réduite, normale, élevée), permettant ainsi aux clients de réduire si nécessaire leur consommation, et aux producteurs de mieux gérer leurs unités de production [8].
- **Lecture automatisée:** les systèmes de reconnaissance des formes permettent de numériser les anciens documents ainsi que les archives, non pas sous la forms d'images, mais plutôt sous une forme textuelle [9].
- **Sécurité:** la reconnaissance vocale et rétinienne sont un exemple d'applications typiques de la reconnaissance des forms pour l'authentification. La vérification des signatures est aussi très populaire [10].

6. SCHÉMA GÉNÉRAL D'UN SYSTÈME DE RECONNAISSANCE DES FORMES

La majorité des systèmes de Reconnaissance des formes ont le schéma de fonctionnement suivant [11].

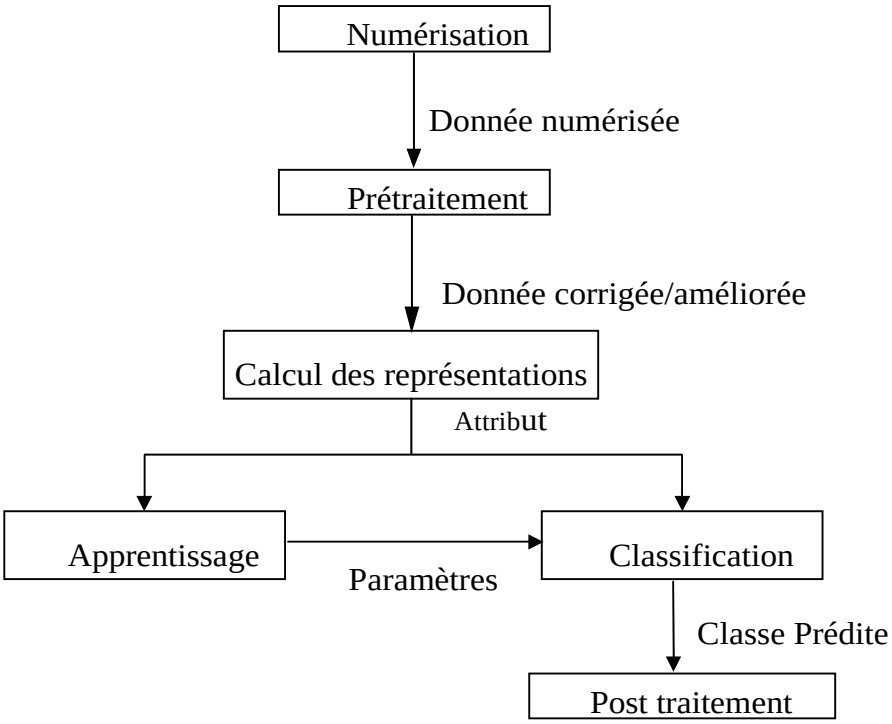


Figure 1.2: Schéma général d'un système de reconnaissance des formes

6.1 PRÉPARATION DES DONNÉES

a. **Numérisation:** À partir des informations du monde physique, construire une représentation des données directement manipulable par la machine. Des capteurs (microphone, caméra, instruments de mesure) convertissent les signaux reçus du monde reel en une representation numérique discrète. L'espace résultant, appelé **espace de représentation** a une dimension **r** très grande lui permettant de disposer du maximum d'informations sur les forms numérisées.

b. **Prétraitement**: Consiste à sélectionner dans l'espace de représentation l'information nécessaire au domaine d'application. Cette sélection passe souvent par l'élimination du bruit, la normalisation des données, ainsi que par la suppression de la redondance. Le nouvel espace de representation a une dimension **r'** très inférieure à **r** mais demeure un espace de grande dimension et contient des informations encore assez

primitives.

c. **Calcul des représentations:** Il s'agit de la phase finale de la préparation des données. Elle fournit un certain nombre de **caractéristiques** ou **paramètres** (les fameux attributs) en utilisant des algorithmes de sélection et/ou d'extraction d'attributs. Les attributs étant limités en nombre, l'**espace des paramètres** ainsi obtenu est de dimension **p** très petite par rapport à **r'**.

6.2 Apprentissage: L'apprentissage ou entraînement, est une partie importante du système de reconnaissance. Le classificateur étant généralement une fonction paramétrique, l'apprentissage va permettre d'optimiser les paramètres du classificateur pour le problème à résoudre, en utilisant des données d'entraînement. Lorsque les données d'entraînement sont préalablement classées, l'apprentissage est dit **supervisé**, sinon il est **non supervisé** [12].

6.3 Classification: cette phase est le noyau de la Reconnaissance des formes. En utilisant les modèles (paramètres) obtenus lors de l'apprentissage, le classificateur assigne à chaque forme inconnue sa ou ses formes les plus probables.

6.4 Post traitement: cette phase a pour but de corriger les résultats de la classification en utilisant des outils spécifiques au domaine d'application. Par exemple pour un système de reconnaissance de textes manuscrits, le classificateur se charge de classer chaque caractère séparément, alors que le post traitement appliqué un correcteur orthographique sur tout le texte pour valider et éventuellement corriger le résultat de la classification. Bien que facultative, cette phase permet d'améliorer considérablement la qualité de la reconnaissance.

7. TRAITEMENT D'IMAGE

L'image fournie par le capteur est transformée en un signal électrique. De ce signal il faut extraire les informations recherchées sur le contenu de la scène dont l'image a été captée.

Les premières bases du traitement d'images sont directement issues du traitement du signal, phénomène normal puisque toute image, qu'elle soit continue ou numérique, peut être considérée comme un signal à 2 dimensions.

La vision n'est pas un domaine facile, car repérer un objet simple dans une image demande beaucoup d'opérations.

L'objectif de ce travail est de montrer comment il est possible de développer des algorithmes de traitement d'images pour réaliser par exemple une segmentation et détection d'objets dans une image.

Le traitement, souvent appelé prétraitement, regroupe toutes les techniques visant à améliorer la qualité d'une image. De ce fait, la donnée de départ est l'image initiale et le résultat est également une image.

La représentation la plus élémentaire correspond à l'image binaire pour laquelle chaque pixel ne peut prendre qu'une valeur parmi deux autres. Pour les images

monochromes, chaque pixel peut prendre une valeur parmi N. N correspond généralement à une puissance de 2, ce qui facilite la représentation de l'image en machine. Par exemple, pour une image en niveau de gris chacun des pixels peut prendre une valeur parmi 256. Sa valeur est alors codée par un octet de donnée. Une image est constituée par une matrice X lignes et Y colonnes de pixels chacun codé par x bits. Les données quantitatives liées à la représentation des images sont représentées dans le tableau suivant (Table 1.3):

1 bit	2 couleurs (noir & blanc)
4 bits	16 couleurs
8 bits	256 couleurs ou niveaux de gris
16 bits	65536 couleurs
24 bits	16777216 couleurs (vraies couleurs)
32 bits	4 294 967 296 couleurs

Figure 1.3: Données quantitatives des images

Dans notre cas l'image est considérée comme une image couleur 320*240 où chaque couleur est codée sur 8 bits. On obtient ainsi environ 786 Ko de données nécessaires pour stocker cette image. La figure 1.4 représente une image correspondant à la représentation RGB (Red , Green Bleu) , on associe un octet à chaque pixel pour définir le niveau d'intensité du pixel la transformation d'une image en niveau de gris, on associe un octet à chaque pixel pour définir le niveau de gris du pixel (255 pour le blanc et 0 pour le noir).

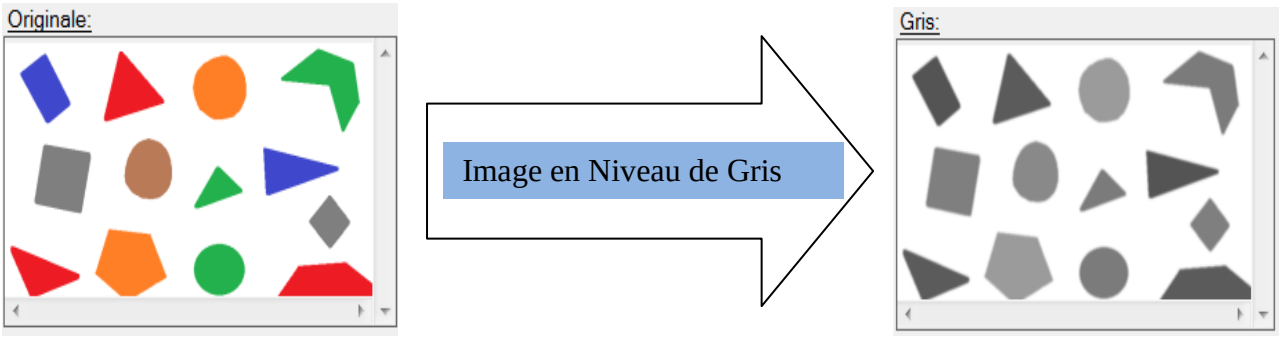


Figure 1.4: Transformation image RGB en image en niveau de gris.

7.1 PRÉTRAITEMENT OU TRAITEMENT BAS NIVEAU

Avant de présenter les différentes étapes de prétraitement, il est nécessaire de définir ce qu'est une image en l'occurrence une image bruitée et le type d'image que nous allons travailler.

7.1.1 DÉFINITION D'IMAGE

Une image est constituée d'un ensemble de points pixels (Picture Element). Il représente le plus petit élément constituant d'une image numérique (on parle d'image numérique lorsque les quantités physiques qui caractérisent l'image sont converties par des valeurs numériques).

L'ensemble des ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

Les axes de l'image sont orientés de la façon suivante:

-L'axe X est orienté à droite (largeur).

-L'axe Y est orienté de haut en bas (hauteur), contrairement aux notations conventionnelles en mathématiques, ou l'axe Y est orienté vers le haut.

Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case est codée par un certain nombre de débits déterminant la couleur ou l'intensité du pixel, la figure 2.3 présente un réseau de pixels (L'élément grisé encadré par la couleur rouge correspond aux coordonnées *i, j*).

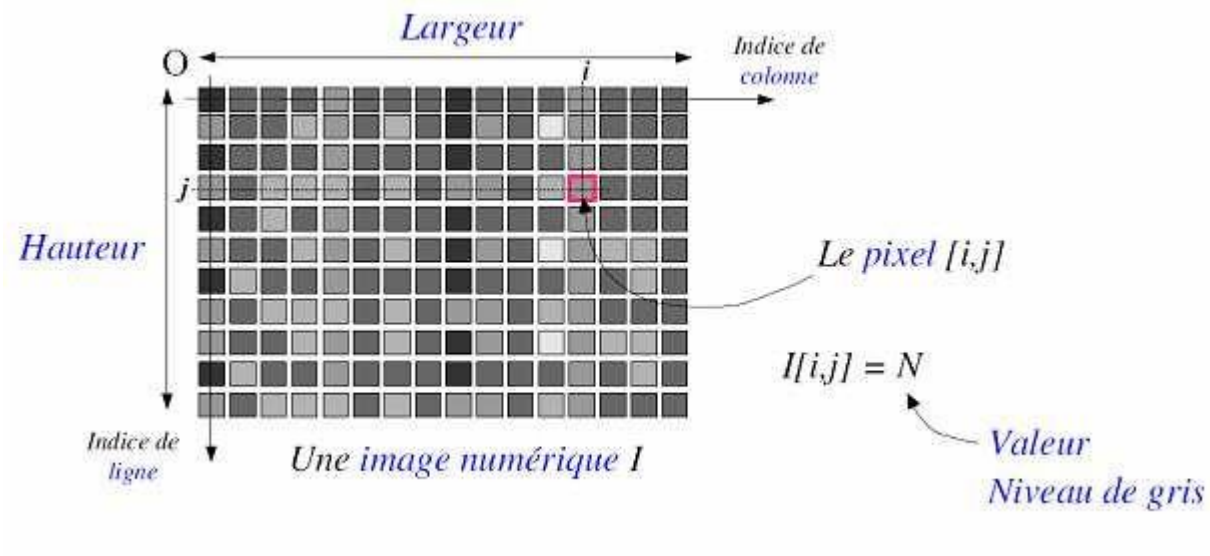


Figure 1.5: Exemple de réseau de pixels

7.1.2 DÉFINITION PIXEL

Le terme pixel est la contraction de l'anglais « Picture element ». C'est le plus petit élément de l'image. Sortie de l'appareil photo une image est composée d'un nombre x de pixels. Un pixel a une couleur exprimée (codée) en langage binaire mathématique (uniquement des 0 et des 1). On comprend donc tout de suite qu'il faut cette quantité minimum de nuances pour avoir une qualité correcte, appelée parfois qualité image.

On dit que l'image est codée en 8 bits par couche (2^8 = 256 niveaux) soit 8 pour le R, 8 pour le V, 8 pour le B au total 24 bits en RVB

7.1.3 IMAGE EN NIVEAU DE GRIS

On passe en niveaux de gris en considérant comme seule composante la luminosité. Il existe plusieurs manières de convertir une image RGB en niveaux de gris.

La plus simple est de faire: gris = $\frac{rouge + vert + bleu}{3}$

Cela équivaut aussi à affecter la couleur en niveau de gris à chacune des trois composantes RGB.

L'idéal est de faire ressortir la luminosité d'un pixel. Celle-ci vient principalement de la présence de la couleur verte. On emploie généralement les coefficients suivants:

$Gris = 0,299 \cdot rouge + 0,587 \cdot vert + 0,114 \cdot bleu$

Dans ce cas on dispose d'une échelle de teintes de gris, et la plupart du temps on dispose de 256 niveaux de gris avec:

0 ----> noir, ...127 ---> gris moyen, ..., 255 ---> blanc

Certaines images peuvent être codées sur deux octets ou plus (certaines images médicales, des images astronomiques,...) ce qui peut poser des problèmes dans la mesure où les systèmes de traitement d'images courants supposent l'utilisation des pixels d'un octet.

Notons au passage que l'humain standard ne reconnaît au plus que 64 niveaux de gris

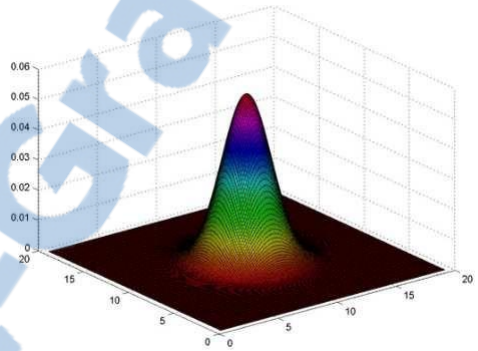
Valeurs des niveaux de gris et teintes de gris correspondantes
0....20....40....50....80....100....120....140....160....180....200....220....240....255

-FILTRE GAUSSIEN

L'intérêt du filtre gaussien est que l'on règle très facilement le degré de filtrage à travers le paramètre σ.On induit une forme de flou qui a pour avantage de retirer les petits défauts. D'un point de vue "signal", ces petits défauts correspondent à des variations importantes d'intensités sur une petite longueur. Si on parlait d'un signal, on dirait que ce sont des "hautes fréquences".

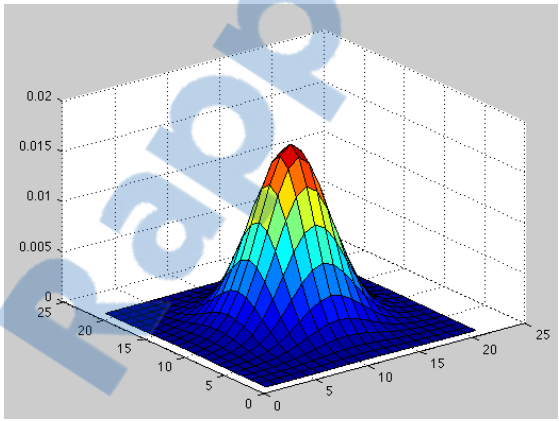
Le paramètre sigma s'appelle la déviation standard, et détermine la largeur de la cloche Gaussienne. Par rapport au filtre moyenneur, le filtre gaussien accorde une grande importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci [13].

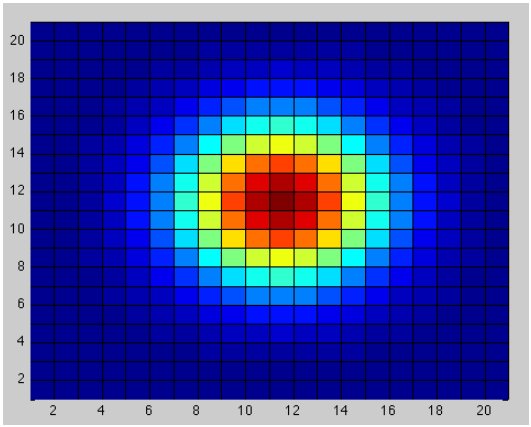
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



-DU CONTINU AU DISCRET

Le noyau gaussien est défini par un ensemble de cœfficients qui sont des échantillons de la Gaussienne 2D





Etant donné que les images numériques sont composées de valeurs discrètes, le noyau gaussien sera composé aussi de valeurs discrètes:

Il faut noter que plus le sigma est grand, plus la cloche Gaussienne est large et plus le flou appliqué à l'image sera marqué.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Pour calculer un flou gaussien, il existe deux principes :

- produit de convolution avec un noyau gaussien
- passage dans l'espace de Fourier

Dans le premier cas : Si l'on veut un flou qui prenne en compte un grand nombre de pixel autour du pixel de destination, il est nécessaire d'augmenter le rayon du noyau (dans le cas d'un produit de convolution standard avec un noyau gaussien). On applique le filtre de bruit gaussien dans la figure 1.6

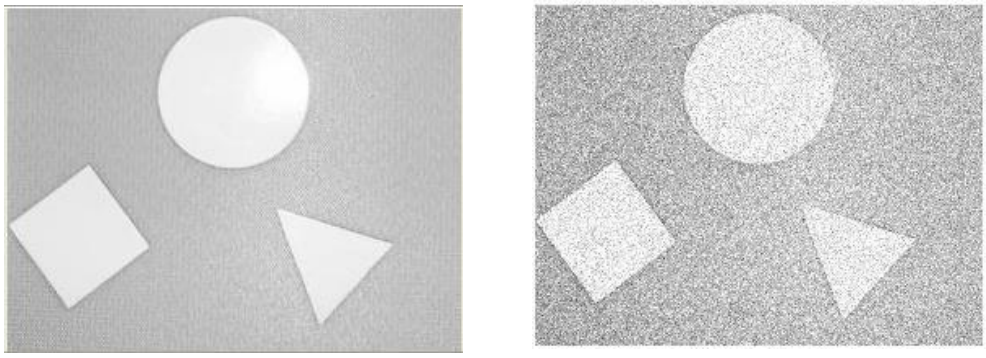


Figure 1.6: Filtre de bruit gaussien

L'avantage de passer dans l'espace de Fourier, c'est que l'on peut appliquer un flou gaussien de n'importe quel rayon (dans ce cas, ce n'est plus vraiment un rayon) pour un même temps d'exécution.

Pour l'appliquer, nous avons un noyau de convolution G et une image I, et G*I le produit de convolution entre G et I.

Alors, si on note F la transformée de Fourier, on a la propriété suivante:

F(G * I) = F(G).F(I) Le produit de convolution (opération lente) se transforme en produit simple (opération rapide). Pour un noyau gaussien, l'avantage étant que F(G) est une formule connue.

Le filtre gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyenneur il préserve mieux l'information contenue dans l'image que les moyenneurs. On considère souvent que le filtre gaussien est le meilleur compromis entre lissage et préservation de l'information, ce qui implique pour un masque de convolution très étendu et donc une charge de calcul enlevée. C'est pourquoi dans la pratique cette solution est rarement retenue et se contente d'un masque peu étendu.

On constate simplement, sur l'exemple de filtrage de notre image que les contours des objets sont réduits, le contraste après filtrage est moins bon.

8. SEGMENTATION DES IMAGES

La segmentation est une étape essentielle en traitement d'images et reste un problème complexe. La segmentation est un processus de la vision par ordinateur, généralement c'est la première étape de l'analyse d'image qui vient après le prétraitement. La segmentation est l'extraction de caractéristiques de l'objet, ce qui permet une distinction entre l'objet et le fond.

Elle aide à localiser et à délimiter les entités présentes dans l'image. Il existe une multitude de méthodes de segmentation dont l'efficacité reste difficile à évaluer.
La segmentation des images est le procédé qui conduit à un découpage de l'image en un nombre fini de régions (ou segments) bien définies qui correspondent à des objets, des parties d'objets ou des groupes d'objets qui apparaissent dans une image. C'est une transformation très utile en vision artificielle.

Une erreur dans la segmentation de la forme à reconnaître augmente forcément le risque d'une mauvaise reconnaissance.
Essentiellement, l'analyse de l'image fait appel à la segmentation où l'on va tenter d'associer à chaque pixel de l'image un label en s'appuyant sur l'information portée (niveaux de gris ou couleur), sa distribution spatiale sur le support image, des modèles simples (le plus souvent des modèles géométriques).

8.1 LES PRINCIPES DE LA SEGMENTATION

De nombreux travaux ont été réalisés sur ce sujet, dans des domaines aussi variés que le domaine médical, militaire, industriel, géophysique, etc… C'est toujours un sujet d'actualité et un problème qui reste ouvert où l'on retrouve de très nombreuses approches:

a. La segmentation par régions
b. La segmentation par seuillage
c. La segmentation par contours
d. La Transformée de Hough
e. La segmentation par étiquetage en composantes connexes
f. La segmentation par LPE

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Toutes ces approches visent à l'extraction des caractéristiques. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale

n'existait pas. Il n'existe pas d'algorithme universel de segmentation à chaque type d'images correspond une approche spécifique. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation.

a. La segmentation par régions

La segmentation va consister à regrouper les pixels de l'image en régions (composantes connexes). Ces régions vérifiant un critère d'homogénéité (par exemple sur les niveaux de gris ou sur la texture...). On cherche par ce traitement à obtenir une description de l'image en régions, elle peut être située dans le cadre plus général de la segmentation dedonnées.

Le problème de base de la segmentation de données est le suivant :

On connaît :

- Un ensemble d'entités.
- Un ensemble d'attributs caractérisant ces entités.
- Des relations topologiques entre ces entités.
- Des attributs relationnels entre ces entitès.

On cherche:

Une (ou des) partition(s) de ces données ayant des propriétés intéressantes par rapport aux attributs et aux relations topologiques.

En général, la segmentation peut être définie comme un partitionnement en régions *Ri* connexes Suivant un prédicat *p* telles que :

$$I = \bigsqcup_{i=1}^{N} R_i$$

R_i est une région connexe

$R_i \sqcap R_j = \varnothing$

$P(R_i) = TRUE$

$P(R_i \sqcup R_j) = FALSE \qquad (i \neq j)$

A chacune de ces régions, doit correspondre un objet dans l'image car, dans ce procédé d'analyse d'images, l'objectif ultime est d'être en mesure de décomposer une image en un groupe d'objets distincts. En général, ces objets ont des propriétés qui leur sont propres par rapport à l'image elle-même. Ainsi, il est possible de distinguer de tels objets par différentes mesures telles:

- Leur aspect connexe
- Leur couleur cohérente

- Leurs contours
- Leur texture
- Des informations apriori

L'intérêt de détecter des régions est de pouvoir les manipuler ensuite pour extraire des caractéristiques de forme, de position, de taille...

b. La segmentation par seuillage

Dans le chapitre précédent, nous avons vu que la distribution des niveaux de gris dans l'image constitue déjà une information précieuse que l'on peut mettre à profit pour identifier des régions particulières de l'image.

Donc le seuillage a pour objectif de segmenter une image en plusieurs classes en n'utilisant que l'histogramme. On suppose donc que l'information associée à l'image permet à elle seule la segmentation, c'est à dire qu'une classe est caractérisée par sa distribution de niveaux de gris.
A chaque pic de l'histogramme est associée une classe.
Nous avons vu qu'il existe de très nombreuses méthodes de seuillage d'un histogramme. La plupart de ces méthodes s'appliquent correctement si l'histogramme contient réellement des pics séparés. De plus, ces méthodes ont très souvent été développées pour traiter le cas particulier de la segmentation en deux classes(bimodale).
L'approche la plus simple est celle où on ne fait intervenir qu'un seuil (on dira que l'image est alors formée d'un fonds et d'un objet). Elle consiste à tester pour chaque pixel de l'image si sa valeur est supérieure ou inférieure à un certain seuil, et produit une image binaire regroupant les résultats.

b.1 Détection automatique du seuil

Il existe plusieurs méthodes pour calculer de manière automatique la valeur du seuil à appliquer. L'une des plus répandue est la méthode d'Otsu. Elle consiste à faire l'hypothèse que l'image contient deux classes, décrites chacune par une partie de l'histogramme de l'image. La qualité du seuillage est quantifiée en mesurant la variance des niveaux de gris de chaque classe. On cherche la valeur de seuil qui minimise les variances des deux classes. La figure 1.7 et la figure 1.8 ci-dessous montrent la binarisation par la méthode OTSU d'un seuil = 136 et d'un seuil = 102

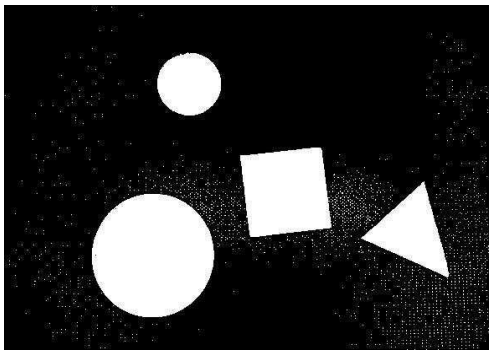


Figure 1.7: La méthode OTSU seuil=136

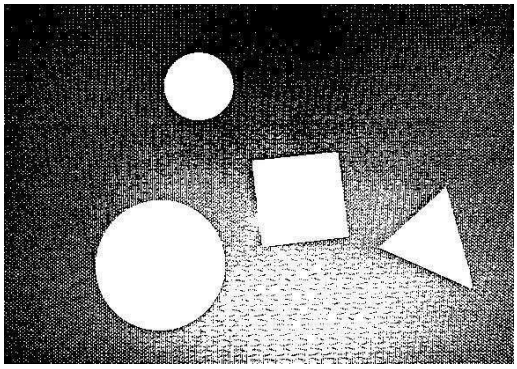


Figure 1.8: La méthode OTSU seuil=102

b.2 Seuillage à hystérésis

Le seuillage à hystérésis est une méthode un peu plus perfectionnée, qui permet d'améliorer la détection des contours des zones contrastées, tout en évitant de détecter des zones du fond. Le principe est d'utiliser deux seuils: un seuil « haut » et un seuil « bas ». On sélectionne d'abord l'ensemble des pixels au dessus du seuil haut, puis l'ensemble des pixels au-dessus du seuil bas. On ne garde ensuite que les composantes connexes du seuil bas qui contiennent au moins un pixel au dessus du seuil haut. En pratique, cette détection peut se faire soit par reconstruction géodésique, soit par étiquetage des composantes connexes de l'image des seuils bas et détection des étiquettes présentes dans l'image des seuils hauts. L'exemple ci-dessous nous montre le seuillage par hystérésis figure 1.9

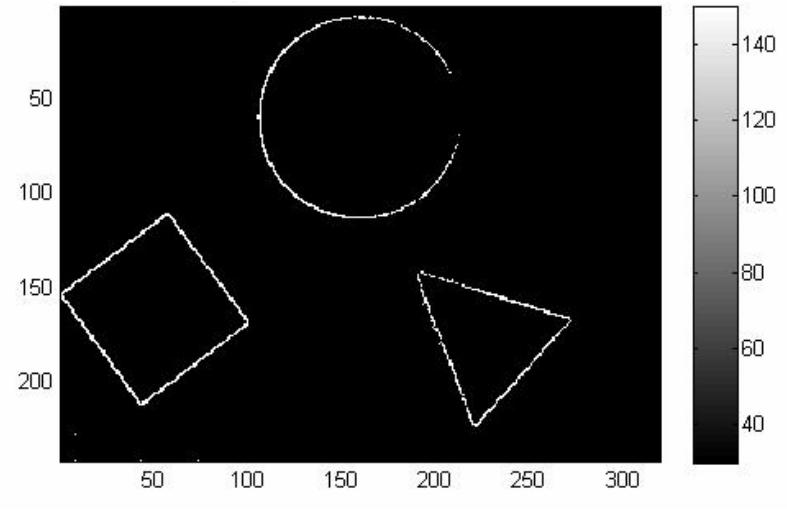


Figure 1.9: Seuillage par hystérésis

c. La segmentation par contours

De nombreux problèmes pratiques d'analyse d'image nécessitent une analyse préalable où il faut découper l'image en régions homogènes séparées par des contours. En général les régions homogènes sont des régions où l'intensité lumineuse varie lentement en fonction des coordonnées spatiales. Les contours séparant ces régions sont des portions de l'image de faible largeur (largeur de l'ordre d'un pixel) où les variations d'intensité sont importantes. Il existe une multitude de méthodes pour détecter les contours, d'où une autre approche de segmentation des images est la recherche des contours des régions. Les différentes méthodes de détection des contours utilisés sont Sobel, Prewitt, Robert, Canny. Ces méthodes ont été proposées pour détecter des transitions dans les images.

c.1 Détection de contours

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. La détection des contours d'une image réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image.

L'extraction de contours, d'un point de vue algorithmique, est une fonction économique qui permet d'alléger considérablement les processus de reconnaissance et localisation d'objet.

Un contour peut être défini comme une "marche d'escalier" si le contour est net, comme une "rampe" si le contour est plus flou ou comme un "toit" s'il s'agit d'une ligne sur un fond uniforme figure 1.10

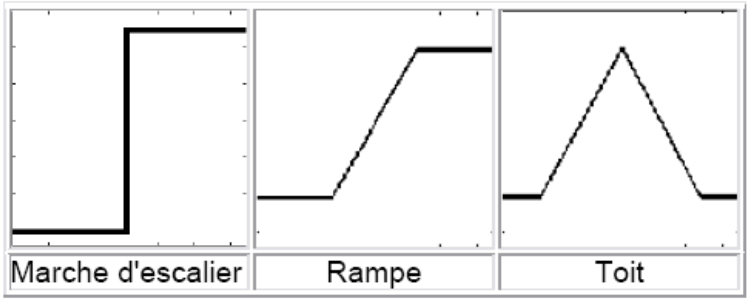


Figure 1.10: Différentes formes de contours

La détection de contours dans notre cas permet de déterminer les limites d'objets pour la segmentation, dont le but est la séparation des objets dans l'image.

c.2 Détection de contours par la méthode de Canny

Cet algorithme a été conçu pour être optimal suivant trois critères :

- Bonne détection : faible taux d'erreur dans la signalisation des contours
- Bonne localisation : minimisation des distances entre les contours détectes et les contours réels
- Clarté de la réponse : une seule réponse par contour et pas de faux positifs.

La détection des contours par Canny se déroule en plusieurs étapes. Tout d'abord il s'agit de réduire le bruit de l'image afin d'optimiser la détection des contours. Ceci permet de réduire de fortes réponses lors du calcul du gradient, conduisant ainsi à de faux positifs. Un filtre Gaussien est utilisé afin de " lisser " l'image originale et donc de réduire le bruit.

Le filtre de Canny est basé sur l'algorithme de Sobel, il dispose d'étapes supplémentaires qui permettent de travailler sur des images plus complexes. Cette méthode permet, en changeant l'intensité de détection de contour, d'éliminer les défauts de linéarisation et du bruit et en même temps d'avoir des contours larges. La largeur est importante pour l'algorithme de localisation par corrélation qui suit cet algorithme de détection de contour. Nous présentons dans ce qui suit les images obtenues par les différentes valeurs de σ avec un seuil = [3e-03 6e-02] ainsi la combinaison des deux valeurs pour différencier la détection des contours

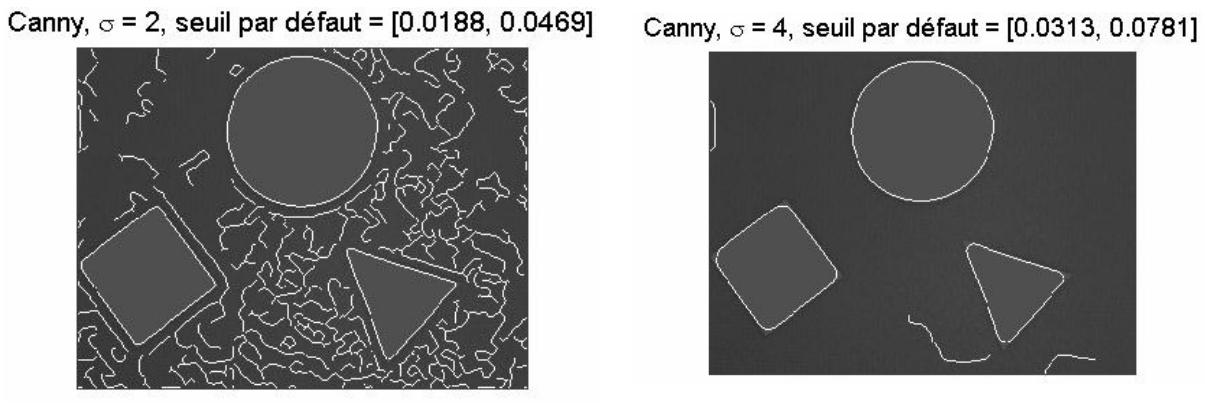
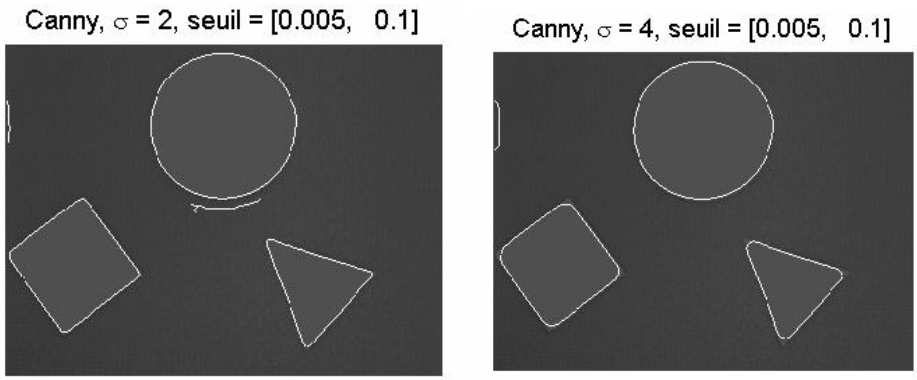


Figure 2.9 : Filtre de Canny



Donc on remarque avec σ= 4 et un seuil = [0.005, 0.1] on obtient un contour fin

d. La transformée de Hough

La transformée de Hough est une technique de reconnaissance de formes inventée en 1962 par Paul Hough. Cette technique nous permet de reconnaître les lignes (droite), les cercles ou n'importe quelle forme présente dans une image [14].

L'application la plus simple permet d'identifier la présence de lignes dans une image. Elle peut être très facilement étendue à la reconnaissance de cercle aux d'autres modèles mathématiques simples. Il existe également une transformée généralisée de Hough.

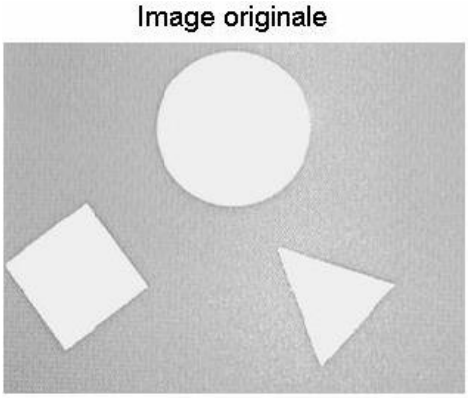
Le Principe est le suivant. Par un point donné peut passer une infinité de droite

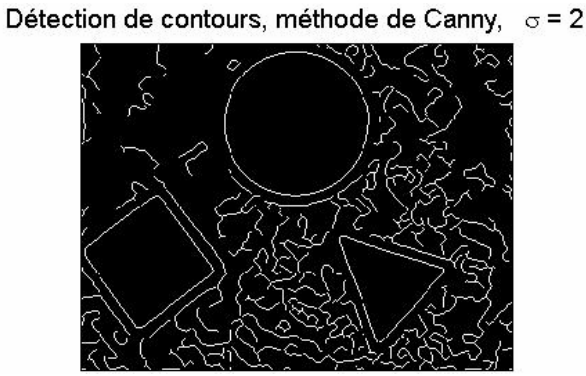
$y=a.x+b$ (ou $r = x \cos\theta + y \sin\theta$ en coordonnées polaires).

Si l'on trace r en fonction de θ, on obtient une sinusoïde représentant l'ensemble des droites de paramètre (r, θ) passant par ce point. C'est ce que l'on appelle « espace de Hough ». Si l'ontrace

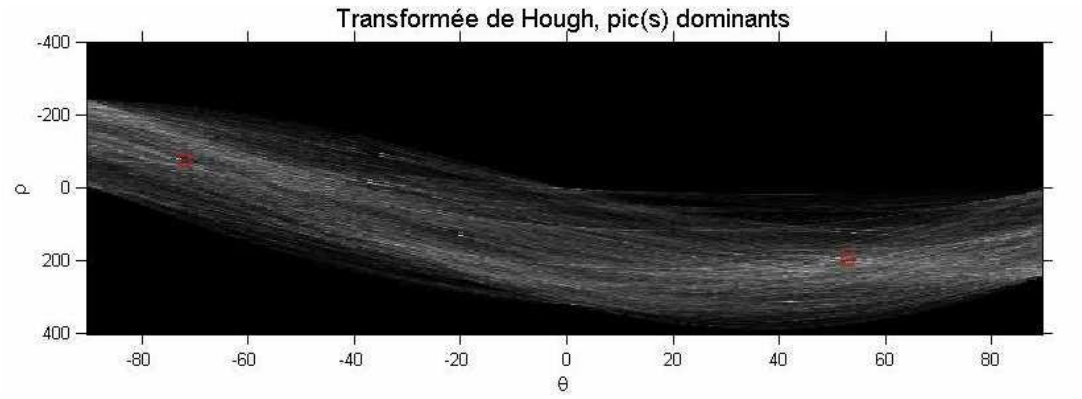
alors une autre sinusoïde par un autre point, l'intersection des deux sinusoïde donne la valeur de (r, θ) de la droite passant par ces deux points. On va donc chercher à tracer pour l'ensemble des points chacune des sinusoïdes correspondantes dans l'espace de Hough. On cherchera alors à savoir quels sont les lieux ou les courbes qui se croisent pour en déduire la présence de droites.

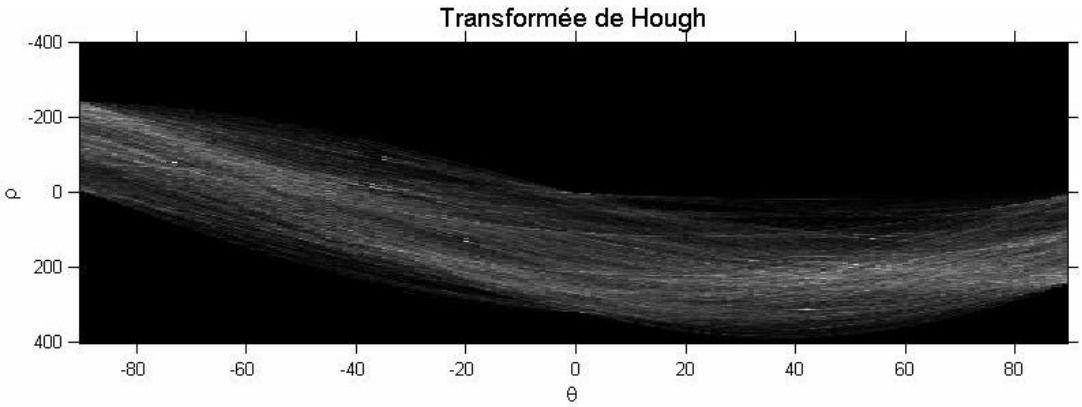
Voici quelques figures de la transformée de Hough par la méthode de canny avec la valeur sigma=2



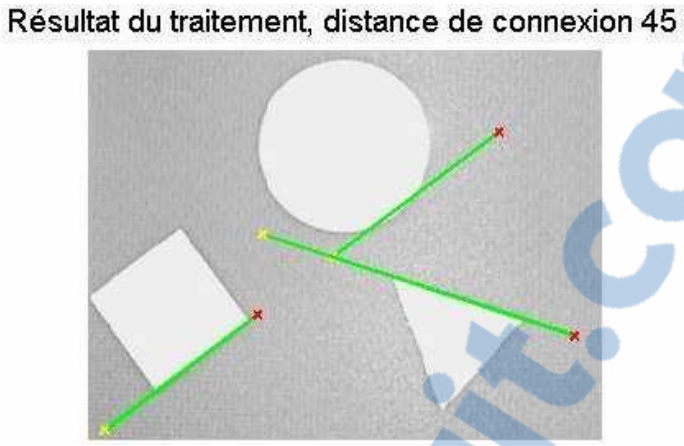


L'image de la transformée de Hough où les 2 pics en carré rouge

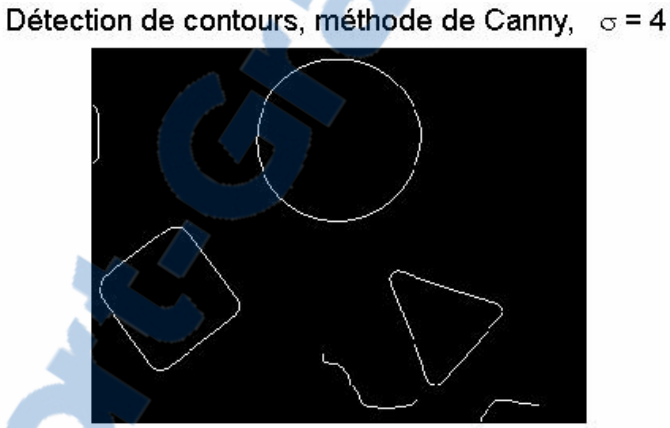


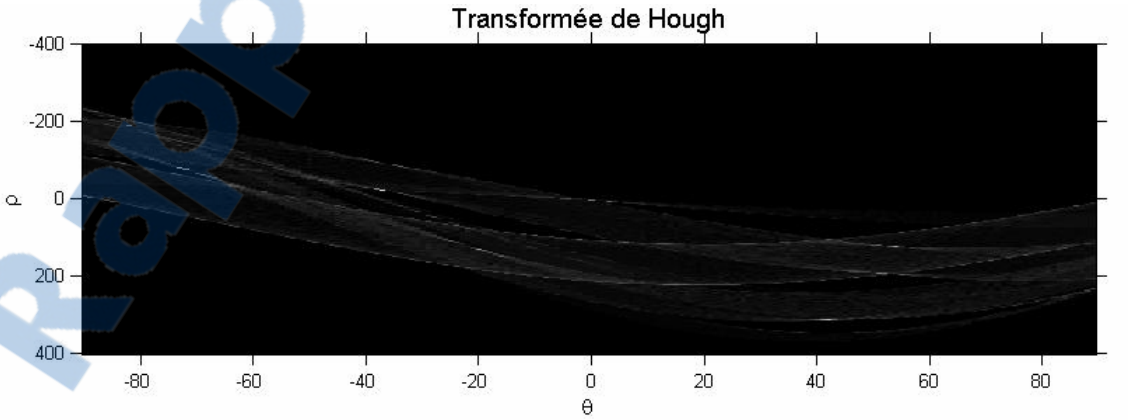


Le résultat du parcours et le tracé des lignes par la transformée de Hough.

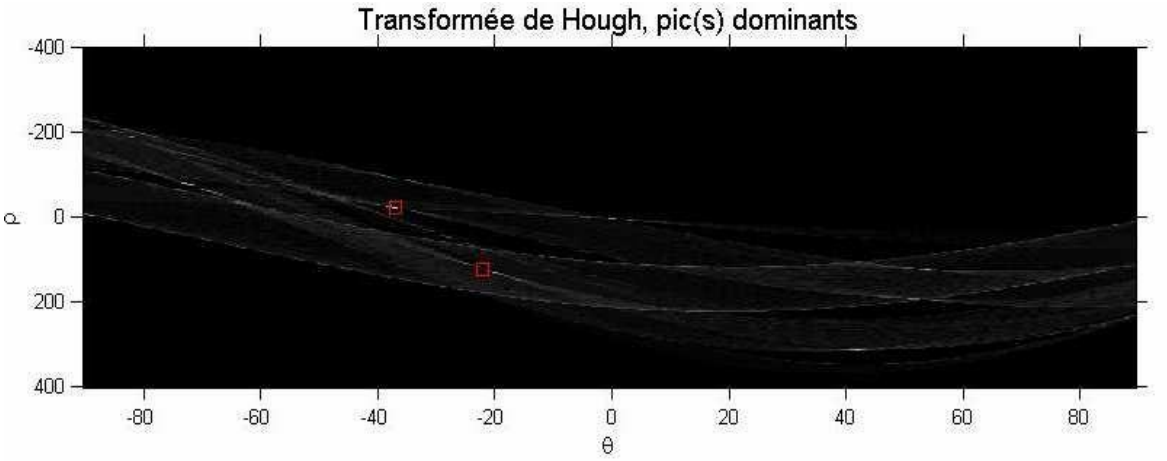


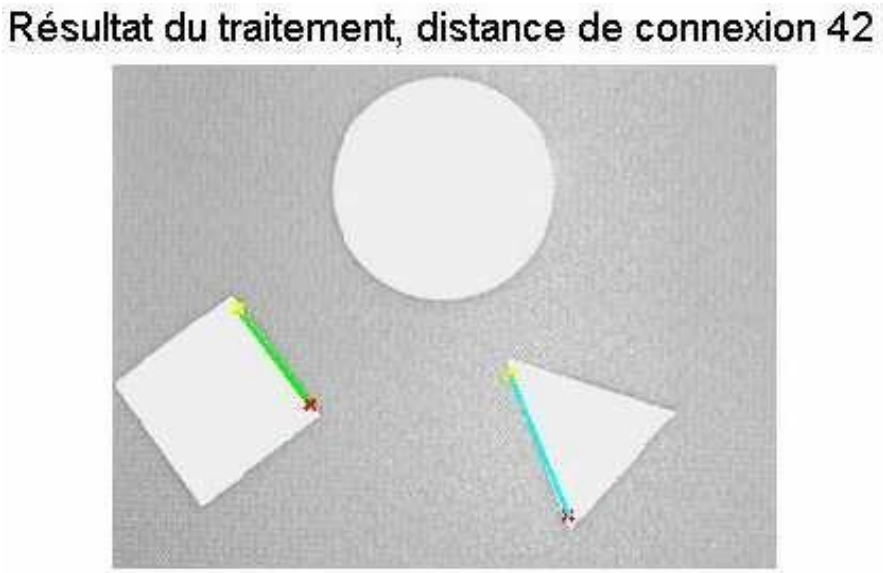
Canny avec sigma =4





RapportGratuit.com





Dans cette partie nous distinguons que la détection de contours est très satisfaisante, puisqu' on a pu avoir les bords extérieurs du carré et le triangle avec la valeur de sigma égale à 4.

Nous avons finalement opté, pour la détection la plus simple, c'est-à-dire simplement en appliquant un filtre de Canny. Car les résultats observés en faisant des prétraitements n'étaient pas convaincants. La détection de contours avec Canny n'est certes pas parfaite mais honorable, et tous les prétraitements appliqués n'apportent aucun plus dans cette détection decontours.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

9. CONCLUSION

La reconnaissance de formes dans les images est un des problèmes les plus difficiles en vision par ordinateur puisqu'il est indispensable pour toutes sortes d'applications.

La vision par ordinateur ne cherche pas à comprendre ou à reproduire la vision humaine, mais à construire un modèle algorithmique qui vu de l'extérieur, possède des propriétés semblables. D'ou la vision artificielle est d'extraire des caractéristiques à partir des images. C'est une mise en correspondence image/reconnaissance.

Connaître la forme nécessite une bonne acquisition. Le choix du type de caméra est très important, l'éclairage est l'aspect le plus critique de toute application de vision artificielle, si l'objet est bien éclairé on peut arriver à de très bons résultats de notre processus de reconnaissance de formes, mais l'éclairage n'est pas le seul paramètre. Il y en a d'autres pour le traitement numérique de l'image, où un appareillage plus complexe, est indispensable.

Ce domaine cherche à détecter la présence de certaines formes, certains contours ou certaines textures d'un modèle connu, c'est le problème de la détection.

Chacune des methods desegmentation a sesavantages, mais a également ses inconvénients. Comme dans la plupart des problèmes, c'est une histoire de compromis.

Les méthodes demandant des paramètres sont très délicates à manipuler. Elles fonctionnent correctement dans le cas où les paramètres sont choisis judicieusement. Ilest évident pour une personne de savoir si la segmentation est correcte, mais pour une machine cela pose réellement de grandes difficultés.

Un aspect important lors du choix de l'une de ces méthodes est le temps d'exécution.

Un bon prétraitement serait envisageable. Il permettrait d'améliorer considérablement le résultat final. Chaque méthode impose de concevoir unprétraitement adapté à celle-ci.

Dans notre travail, nous avons utilisé la technique de seuillage et la détermination des contours de Canny car incorporée dans la bibliothèque OPENCV utilisée pour reconnaitre les formes et d'après la littérature c'est la technique qui conviendrait le plus à ce que nos souhaitons.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Chapitre2

Chapitre 2: *Mise en place d'une application de Reconnaissance de Formes*

1. INTRODUCTION

Ce dernier chapitre est consacré à la conception de l'application qui permettra d'identifier des Objets par reconnaissance de formes en utilisant deux methodes la capture par la WebCam ou par fichier Image. Le programme de reconnaissance de formes a été écrit en Visual Basic. Il a fait appel aux bibliothèques OpenCV et le logiciel Visual Studio 2010 pour pouvoir faire des traitements aisément tels que prendre des photos en temps réel à l'aide de la webcam d'un ordinateur, faire le traitement des formes.

2. PRESENTATION DE LA LIBRAIRIE OPENCV

2.1 HISTORIQUE

Officiellement lancé en 1999, le projet OpenCV est développé initialement par Intel pouroptimiser les applications gourmandes en temps processeur. Cela faisait partie d'une série deprojets tels que l'affichage d'un mur en 3 dimensions. Cette bibliothèque est distribuée souslicence BSD.

Les principaux acteurs du projet sont l'équipe de développement de bibliothèque de chez Intelainsi qu'un certain nombre d'experts dans l'optimisation de chez Intel Russie.

Les objectifs de base du projet étaient :

- Faire des recherches sur la vision par ordinateur en vue de fournir un logiciel libre etoptimisé.
- Établir une infrastructure commune s'appuyant sur les développeurs pour obtenir uncode plus lisible et transférable.
- Continuer à développer en rendant le code portable et permettre des performancesoptimisées gratuites avec une licence qui est libre de toutes contraintes commerciales.

La première version alpha d'OpenCV fut présentée lors de la conférence IEEE sur la visionpar ordinateur et la reconnaissance de formes en 2000. Après cela, cinq versions bêta ont étépubliées entre 2001 et 2005 et la première version 1.0 a été publiée en 2006.

Au milieu de l'année 2008, OpenCV obtient l'appui de la société de robotique Willow Garageet la bibliothèque est encore développée à ce jour. Une version 1.1 est sortie en Octobre 2008et un livre écrit par deux auteurs d'OpenCV, publié par O'Reilly Media est sorti sur le marchéce même mois.

La deuxième version majeure d'OpenCV née en octobre 2009. Il s'agit d'OpenCV 2 incluantdes changements majeurs au niveau du langage C++ servant à faciliter le développement denouvelles fonctions et améliorant les performances. [15]

2.2 QU'EST-CE QUE OPENCV

OpenCV (Open Source Computer Vision) est une bibliothèque libre de vision parordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windowset Mac OS X. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage.Open CV est orienté vers des applications en temps réel.Un des buts d'OpenCV est d'aider les gens à construire rapidement des applicationssophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur. Labibliothèque d'OpenCV contient plus de 500 fonctions. [16] [17]

2.3 FONCTIONNALITES

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiéespermettant de créer des programmes partant des données brutes pour aller jusqu'à la créationd'interfaces graphiques basiques. Elle propose la plupart des opérations classiques entraitement bas niveau des images et des vidéos.

- **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.
- lissage, filtrage.
- seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- segmentation (composantes connexes, GrabCut).
- morphologie mathématique.

- **Traitement vidéo**

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherchepparce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision desordinateurs 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.
- cascade de classifieurs boostés.

- détection de mouvement, historique du mouvement.
- poursuite d'objets par mean-shiftou Camshift.
- détection de points d'intérêts.
- estimation de flux optique (Méthode de Lucas–Kanade).
- triangulation de Delaunay.
- diagramme de Voronoi.
- enveloppe convexe.
- ajustement d'une ellipse à un ensemble de points par la méthode des moindrescarrés.

2.4 MODULES OPENCV

OpenCV a une structure modulaire. Les principaux modules d'OpenCV sont énumérés ci-dessous :

- **CxCore (**Coeur d'OpenCV) :

Contient les structures de donnée et les opérations mathématiques de base. Elle permet deréaliser les fonctions d'algèbre matriciel, la transformation de données, la gestion de mémoire,la manipulation d'erreurs, le chargement dynamique de code et permet aussi la réalisation desgraphiques.

- **HighGui (librairie d'interfaçage graphique)**

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

- **ImgProc (traitement d'image) :**

Ce module comprend des algorithmes de traitement d'image de base, y compris le filtraged'image, transformations d'image, les conversions d'espace couleur et etc.

- **Vidéo (traitement de flux vidéo)**

Ceci est une vidéo module d'analyse qui inclut des algorithmes de suivi (tracking) d'objets,des algorithmes de soustraction de fond et etc.

- **ObjDetect (détection d'objets)**

Cela comprend l'objet de détection et de reconnaissance des algorithmes pour les objetsstandards.

- **ML (librairie d'apprentissage automatique)**

Comporte les fonctions de classification, d'analyse de donnée et des outils de clustering.

- **Calib3D**

Calibrage caméra et reconstruction 3D. [18] [16]

2.5 MICROSOFT VISUAL STUDIO 2010

Microsoft Visual Studio est une suite de logiciels de développement pour Windows et macOS conçue par Microsoft. La dernière version s'appelle **Visual Studio 2017**.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications webASP.NET, des services webXML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

Durant sa conférence *Connect()* 2016, Microsoft a annoncé le portage de Visual Studio sur macOS, le système d'exploitation d'Apple.

3 ENVIRONNEMENT DU TRAVAIL

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

3.1 ENVIRONNEMENT MATERIEL OU LE HARDWARE:

Un ordinateur ASUS avec les caractéristiques suivantes:

_ Processeur: Intel (R) Core (TM) i3-2370M CPU @ 2.40 GHz 2.40GHz

_ RAM: 4.00 GO.

_ Disque Dur: 500 GO.

_ Webcam intégréeUSB 2.0 UVC VGAWebCam Camera.

3.2 ENVIRONNEMENT IMMATERIEL OU LE SOFTWARE :

_ Système d'exploitation : Microsoft Windows 7 - 64 bits

_ Logiciel : Microsoft Visual Studio version 2010

_ La bibliothèque : OPENCV version 2.9 et v3.1

4 DIAGRAMME DE CLASSES

Le **diagramme de classes** est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un *moule à objet* : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

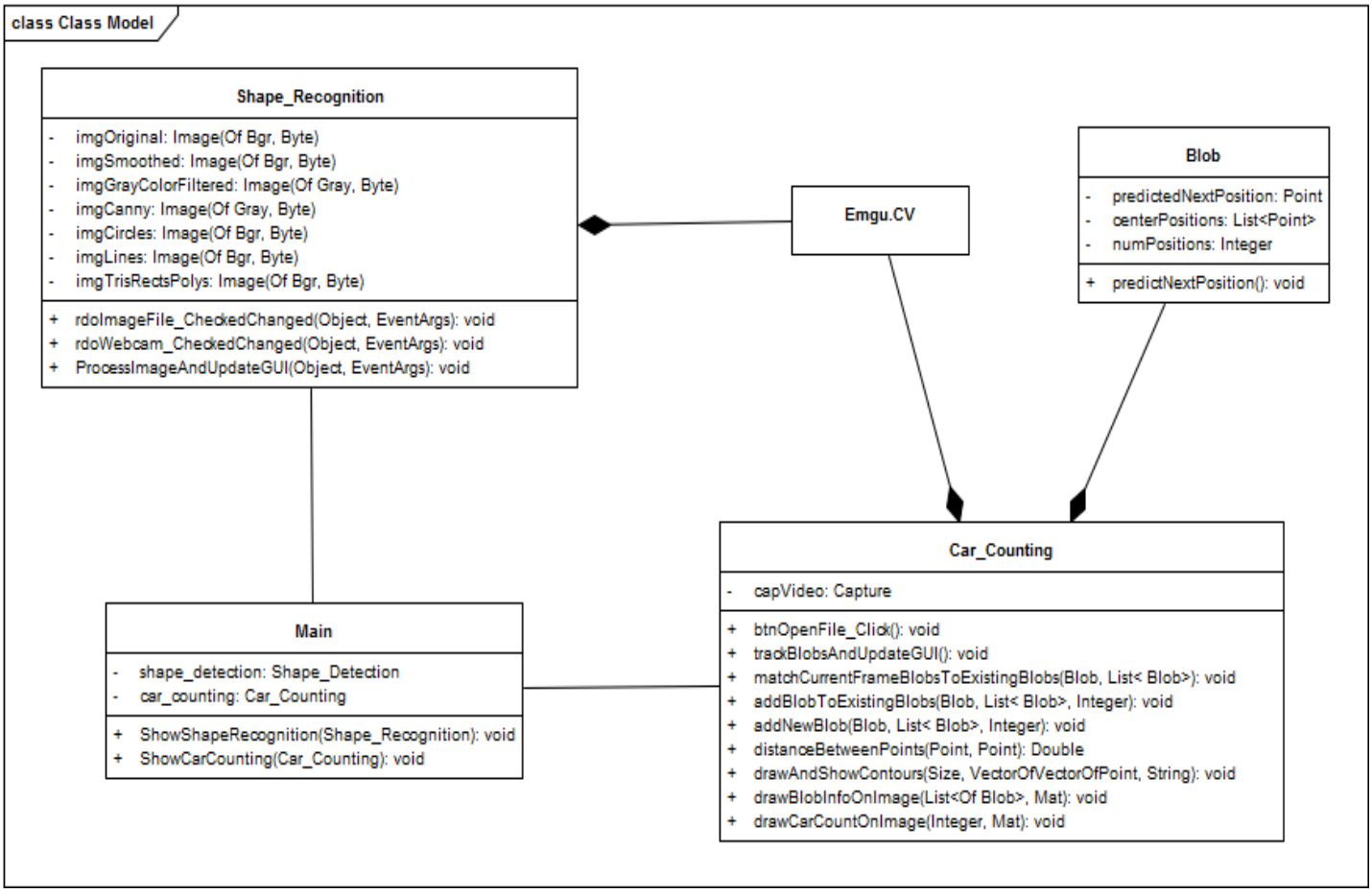


Figure 2.1: Diagramme de Classe en UML

5 INTERFACES DE L'APPLICATION

- C'est l'interface primcipale de notre application, elle comtient deux boutons le premier Reconnaissance de Formes et le deuxième celui de Comtage d'Automobiles

Figure 2.2: Interface d'accueil de l'application Reconnaissance de Formes

- Cette interface présente le menu principal qui permet une sorte de passerelle pour accéder à l'application de reconnaissance de formes par la selection du fichier image.

Figure 2.3: L'interface de notre application par la selection Fichier d'image.

Rapport-Gratuit.com

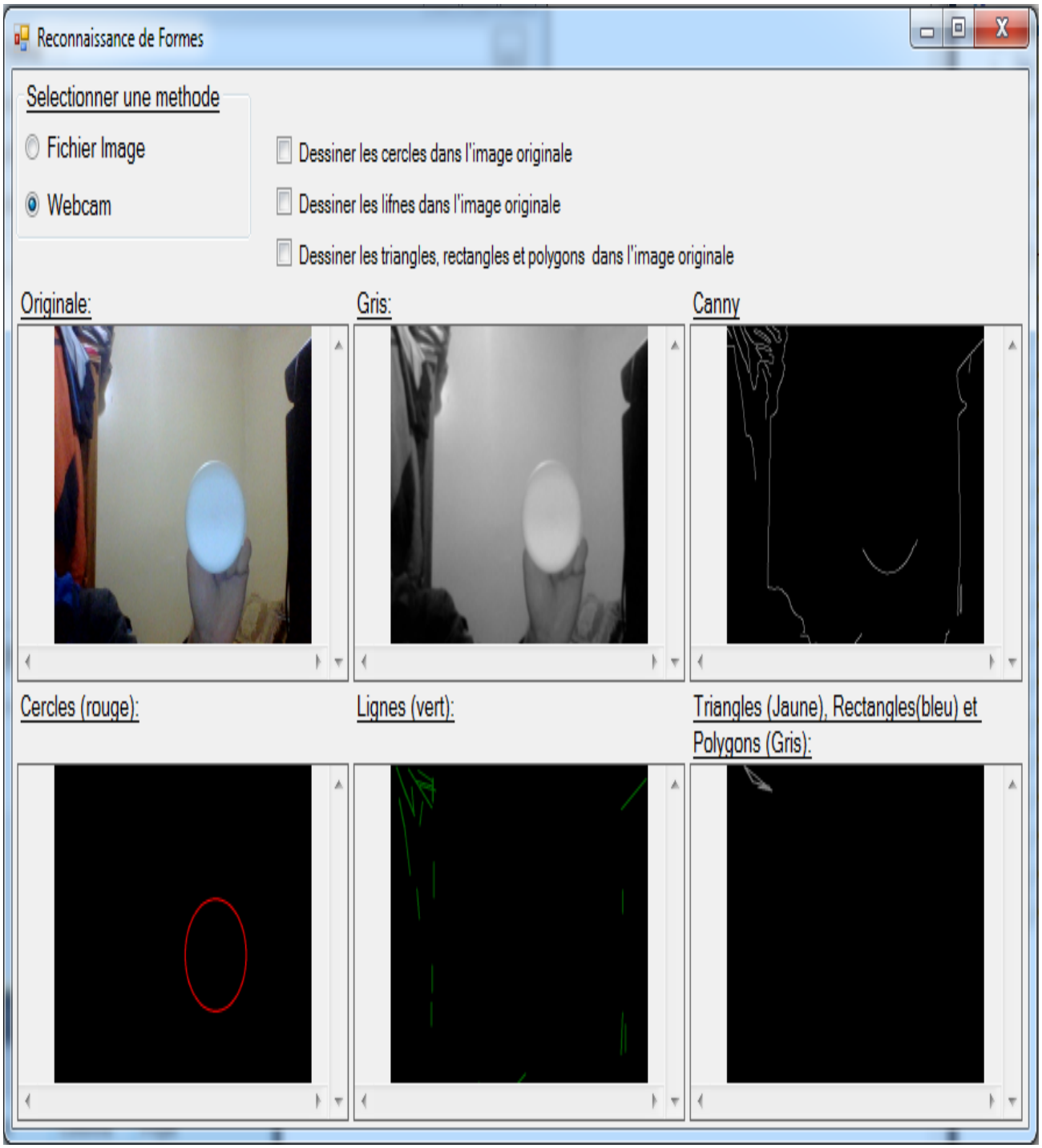


Figure 2.4: L'interface de notre application par la selection WebCam.

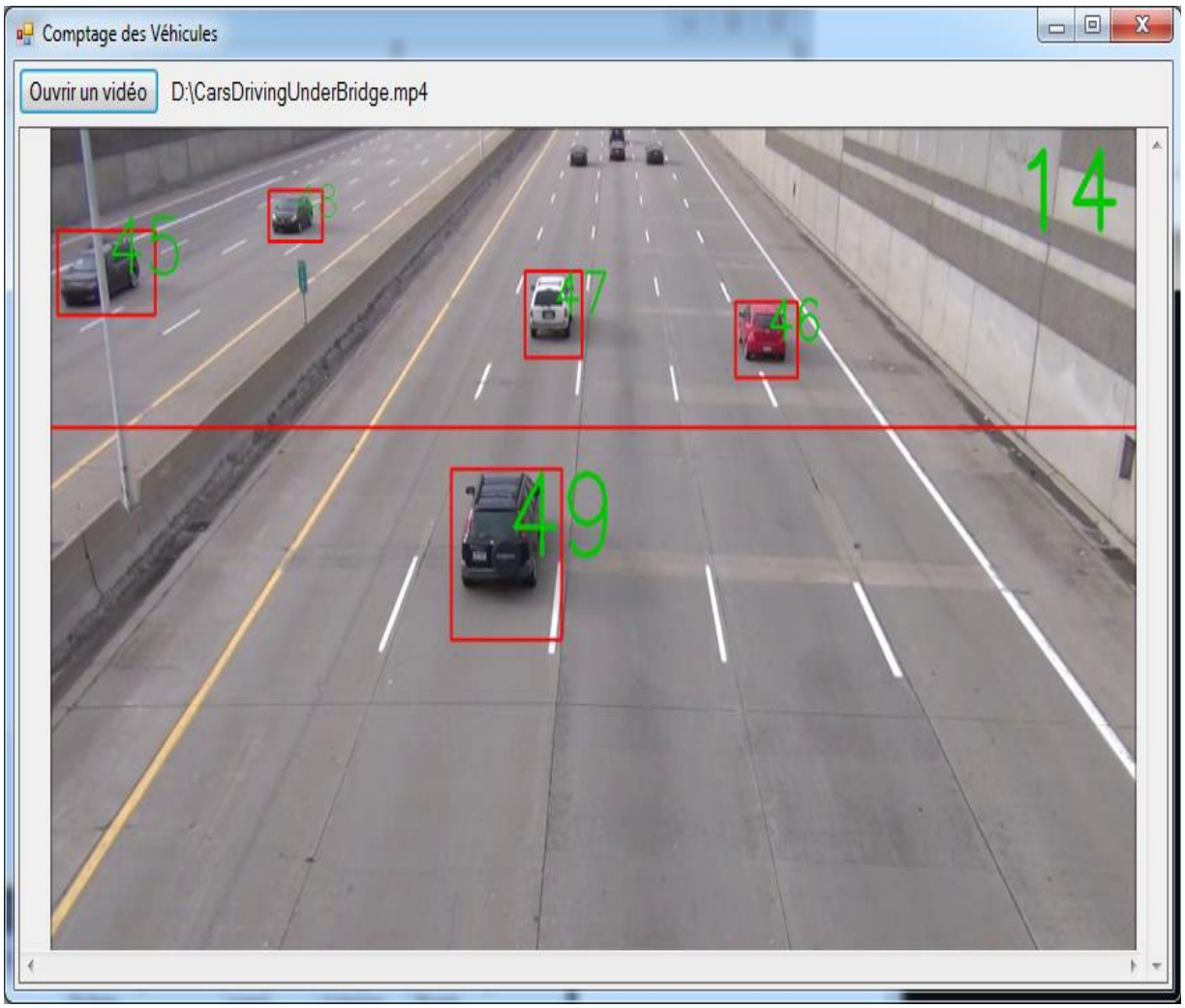


Figure 2.5: Détection de la forme d'objets en mouvement.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

6 CONCLUSION

Nous avons réussi à mettre au point un système de reconnaissance de formes via des images statiques, une image capturée par une web cam et enfin une image défilante sur ue vidéo. Ce système permet de faire les traitements des images sous formes géométriques et de les traiter pour faire la selection des formes (Triangle, carré, Cercle, Polygone…) avec des couleurs différentes.

Les résultats obtenus démontrent que notre système n'est pas fiable à 100% à cause des problèmes d'éclairage mais nous avons atteint nos objectifs de reconnaissance de formes.

Ce PFE nouus a permis de mieux cerner la thématique de la reconnaissance des formes qui passe par le traitement d'images, qui est comme tout monde le sait en perpétuel changement, ou la recherche y est très active. De nouvelles techniques apparaissent réglièrement, essayant toutes de faciliter la vision par ordinateur, pas encore aboutie.

CONCLUSION GENERALE

La Vision par Ordinateur est à la base de tout système de vision artificielle qui prend en entrée une ou plusieurs images numériques, et qui effectue des traitements sur ces images afin d'en extraire des informations relatives à la scène observée.

La reconnaissance d'objet dans les images est une discipline qui, bien qu'elle soit encore dans une phase de recherche et de perfectionnement, est utilisée par de nombreuses applications. Elle permet à un ordinateur de « comprendre » l'image d'où un lien étroit avec l'intelligence artificielle.

Dans ce projet, nous avons tenté de mettre au point un système de reconnaissance de formes dans une image pour une application de tri et d'attributiond'une couleur à chaque forme. L'objectif est la recherche d'objets dans des images: il s'agira d'image unitaire et non pas de séquences d'images.

Ce projet a été vraiment très intéressant car cela nous a permis d'approfondir et mettre en pratique nos connaissances dans le domaine du traitement d'images mais aussi d'être confronté aux difficultés des images réelles: qualité, bruit, résolution,etc…

Il nous a fallu pas mal de temps pour rechercher les documents nous permettant de mener à terme ce projet, surtout en ce qui concerne l'identification et la localisation des objets dans l'image , le sujet est certes beaucoup trop vaste mais incontournable .Nous sommes confrontés à des difficultés d'acquisition ; puisque nous avons utilisé une simple caméra Webcam qui n'est pas destinée à l'usage professionnel, ainsi que l'éclairage qui est l'aspect le plus critique de toute application de vision artificielle, car si l'image est bien éclairée le processus de reconnaissance d'objets peut atteindre de très bonrésultats.

L'acquisition d'image, nous permet de récupérer les données d'une image en temps réel tout en visualisant l'image, cette fonction nous donne en retour l'image que l'on exploite sous forme matricielle. Celle-ci montre trois matrices représentant l'image dans les couleurs de base. Notre travail est justifié par une transformation d'image en niveau de gris. Nous avons analysé quelques points très importants dans la reconnaissance de formes, de la façon d'appliquer toutes sortes de méthodes de filtrage pour éliminer le bruit d'image, nous avons mis une méthode pour la segmentation d'images sur un fond blanc, les objets en noir,

c'est une étape de binarisation, nous avons introduit l'approche corrélation qui est une méthode efficace et rapide pour la localisation de certaines formes géométriques. Les méthodes de reconnaissance de formes proposées par Hough sont des méthodes d'étude morphologique, mais elles nécessitent un temps de calcul important dans le cas des formes géométriques complexes ou descontours discontinus (Freeman), la méthode de détection de contour de Canny nous a permis de ressortir des contours assez larges qui conviennent parfaitement à la localisation parcorrélation.

La bibliothèque OpenCV est un élément important dans la réalisation et le développement de notre projet et cela est surtout du à la richesse de différentes fonctions qui le compose surtout dans le traitement des images et des vidéos.
Chacun des algorithmes étudiés est plus efficaces que les autres dans des cas précis, du fait de leur simplicité ou de leur précision. Ils ont néanmoins un point commun, il est nécessaire de faire des compromis, il est nécessair de reconnaitre l'objet à chaque instant, mais sans commettre d'erreurs (fausse détection).

Comme perspective de notre travail, nous souhaitons pour les prochains projets de fin d'étude, qu'ils élargissent notre travail de détection et suivre un objet en mouvement mais selon différentes formes géométriques plus complexes.

References bibliographiques

[1] **Watanabe,** S. Watanabe, “Pattern recognition: Human and Mechanical”, New York Wiley, 1985

[2] **Theodoridis et Koutroumbas,** S. Theodoridis, K. Koutroumbas, “Pattern Recognition, Second Edition”, Academic *Press, Elsevier*, 2003.

[3] https://interstices.info/jcms/p_88807/marvin-minsky-un-des-cerveaux-de-l-intelligence-artificielle

[4] V. Bjorn, “One Finger at a Time: Best Practices for Biometric Security,” Banking Information Source (Document ID: 1697301411), April, 2009 .

[5] **Chung et al,** K.W. Cheung, J.T. Kwok, M.H. Law, K.C. Tsui,"Mining customer product ratings for personalized marketing", *Decision Support Systems*, vol. 35, issue 2, pp. 231-243, 2003

[6] **Kaastra et Boyd,** I. Kaastra, M. Boyd, "Designing a neural network for forecasting financial and economic time series", *Neurocomputing*. Vol. 10, no. 3, pp. 215-236. 1996.

[7] **Gupta et al,** S.K. Gupta, W.C. Regli, D.S. Nau, "Manufacturing Feature Instances: Which Ones to Recognize?", *Symposium on Solid Modeling and Applications*, pp. 141-152, 1995.

[8] **Hippert et al,** H.S. Hippert, C.E. Pedreira, R.C. Souza, "Neural networks for short-term load forecasting: a review and evaluation", *IEEE Transactions on Power Systems*, vol. 16, Part 1, pp. 44-55, 2001.

[9] **Munich et Perona,** M.E. Munich, P. Perona, "Visual Input for Pen-Based Computers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, issue 3, 2002.

[10] **Yang et al,** M.H. Yang, D.J. Kriegman, N. Ahuja, "Detecting Faces in Images: A

Survey", *IEEE Transactions On Pattern Analysis And Machine Intelligence PAMI*, vol.24; part 1, pp. 34-58, 2002.

[11] **Woznica et Menal,** P. Woznica, M. Mennal, "Reconnaissance des formes", 2003.

[12] **Theodoridis et Koutroumbas,** S. Theodoridis, K. Koutroumbas, "Pattern Recognition, Second Edition",
Academic Press, Elsevier, 2003.

[13] **J.P. Cocquerez et S. Philipp**, Analyse d'images : filtrage et segmentation Ed. Masson, Année 1995.

[14] **Phan Viet Anhe,** Développement d'un module de segmentation pour un système de reconnaissance biométrique basé sur l'iris, novembre 2008.

[15] Samarth Brahmbhatt, "Practical OpenCV", 2013.

[16] G.BRADSKI & A.KAELHER, "Learning OpenCV Computer Vision with the OpenCV Library" O'Reilly, Sebastopol, 2008.

[17]Samuel Rouxel, " La bibliothèque de traitement d'images OPENCV ", CRESITTIndustrie, 2010.

[18] http://master-ivi.univ-lille1.fr/fichiers/Cours/pje-semaine-3-opencv.pdf

[19] https://www.interstices.info/jcms/c_5952/histoire-du-traitement-d-images