

Sommaire

Introduction Générale

Chapitre 1 Généralités sur la reconnaissance des gestes de la main.....	6
I. Introduction	6
1.1. LES GESTES DE LA MAIN	7
1.2. TAXONOMIE DES GESTES DE QUEK	8
1.3. GESTES STATIQUES ET DYNAMIQUES	8
II. Etat de l’art « Interaction Homme – Machine ».....	9
2.1. DISPOSITIFS D’INTERACTION	9
2.1.1. PERIPHERIQUES D’ENTREE.....	9
2.1.2. LES GANTS DE DONNEES	9
2.1.3. ÉCRANS TACTILES	10
2.1.4. LES CAMERAS VIDEO	10
2.2. APPLICATION ET NOUVELLES POSSIBILITES D’INTERACTION	11
2.2.1. RECONNAISSANCE DE LA LANGUE DES SIGNES	11
2.2.2. REALITE VIRTUELLE	11
2.2.3. REALITE AUGMENTEE	12
2.2.4. CONTROLE MACHINE :	12
III. Les Difficultés liées à la vision par ordinateur.....	13
IV. Conclusion.....	13
Chapitre 2 Système de Reconnaissance des gestes de la main.....	15
I. Introduction	15
II. Système de reconnaissance des gestes de la main.....	15
2.1. TABLETTES GRAPHIQUES & ECRANS TACTILE	15
2.2. PORTABLE DISPOSITIFS ELECTROMAGNETIQUES, GANTS NUMERIQUE :	16
2.3. LA VISION PAR ORDINATEUR :.....	16
2.3.1. Qu’est ce que la vision	16
2.3.2. Pourquoi la vision	17
2.3.3. Capteurs d’acquisition :	17
III. La segmentation	17
3.1. LA SEGMENTATION PAR DETECTION DE LA COULEUR DE PEAU	18
3.1.1. RGB	18
3.1.2. YCbCr chrominance de luminance	18
3.1.3. HSV Teinte, Saturation et valeur	18
IV. Seuillage (thresholding)	19

V.	Contours	20
5.1.	FILTRES DE DETECTION DE CONTOURS :	20
5.1.1.	Approche gradient :	20
5.1.2.	Approches Laplacien :	21
VI.	Codage.....	21
6.1.	CODAGE DE FREEMAN	21
VII.	Classification.....	22
VIII.	Conclusion.....	22
Chapitre 3 Implémentation & Résultats		24
I.	Introduction	24
II.	Environnement de développement.....	24
III.	Architecture matériel	25
IV.	Système de reconnaissance du geste.....	25
4.1.	ACQUISITION	25
4.2.	SEGMENTATION	26
4.2.1.	RGB :	26
4.2.2.	Méthode de seuillage :	27
4.3.	EXTRACTION DE CONTOUR ET DES CARACTERISTIQUES	27
4.3.1.	DESCRIPTION DE LA REGION DE LA MAIN	27
4.3.2.	LES CARACTERISTIQUES GEOMETRIQUES	28
V.	EXPÉRIENCES ET RÉSULTATS.....	29
VI.	Avantage de l'application	35
VII.	Mode d'utilisation.....	35
VIII.	Conclusion.....	35

Liste des Figures

Figure 1 - WIMP "Fenêtre, Icône, Menu, Pointers"	6
Figure 2 - Reconnaissance des gestes avec les gants	6
Figure 3 - Taxonomie des gestes de Quek	8
Figure 4 - "Exemple de gestes statiques & Dynamique"	8
Figure 5 - (a) Trackball – (b) la Wiimote de Nintendo	9
Figure 6 - Les Gants de données.....	10
Figure 7 - (a) Geste de commande – (b) Alphabets simplifié.....	10
Figure 8 - Réalité Virtuelle	11
Figure 9 - Réalité augmentée.....	12
Figure 10 - Souris gestuelle	12
Figure 11 - Schéma général d'un système de reconnaissance des gestes	15
Figure 12 - Ecran Tactile & Tablette graphique.....	16
Figure 13 - Gant Numérique	16
Figure 14 - Théorie de la vision	17
Figure 15 - Exemple de Segmentation par seuillage	19
Figure 16 - Etapes de l'Algorithme de reconnaissance.....	24
Figure 17 - Les Trois types de gestes	25
Figure 18 - Interface	29
Figure 19 - Choix de la Webcam.....	30
Figure 20 - Bouton « Démarrer & Arrêter ».....	30
Figure 21 - Choix du Seuil	31
Figure 22 - Images Capturées par la Webcam	31
Figure 23 - Cadre capturé & traité.....	32
Figure 24 - Geste "STOP"	32
Figure 25 - Geste « Droit ».....	33
Figure 26 - Geste « Gauche »	33
Figure 27 - "Pas de Geste"	34
Figure 28 - Zone 1 « Etape Extraction du contour ».....	34

Introduction Générale

Le sujet de notre projet de fin d'études concerne le développement de méthodes de vision par ordinateur pour la reconnaissance de gestes de la main.

Les gestes de la main sont un canal de communication naturel et intuitif chez l'homme pour interagir avec son environnement. Ils servent à désigner ou à manipuler des objets, à renforcer la parole, ou à communiquer de manière basique dans un environnement bruité. Ils peuvent aussi représenter un langage à part entière avec la langue des signes. Les gestes peuvent avoir une signification différente suivant la langue ou la culture : les langues des signes en particulier sont spécifiques à chaque langue. Depuis déjà plusieurs années l'homme essaye de doter la machine de la vision et ce pour en faire un allié. Mais la vérité est que c'est une tâche très ardue, et même les machines les plus puissantes avec les algorithmes les plus pertinents n'ont pas encore permis de régler cette problématique de manière satisfaisante. En attendant, les quelques résultats obtenus nous ont permis de répondre à certaines utilisations de manière fort satisfaisante. Cependant les recherches dans ce domaine continuent et ne cessent d'améliorer les solutions apportées à la vision machine, avec l'augmentation de la puissance des ordinateurs (processeurs, ram et algorithmique, etc), mais la vision comme nous la pratiquons nous les humains, est encore utopique.

Dans notre projet, on s'est posé comme objectif de développer une application qui reconnaitra certains gestes de la main comme une première étape dans la vision par ordinateur. Pour se faire notre manuscrit est partagé en trois chapitres comme suit :

- ✓ Chapitre 1 : présente le domaine de l'interaction homme-machine gestuelle. Les dispositifs classiques d'interaction sont étudiés, ainsi que les nouvelles possibilités d'interaction. Les nombreuses applications sont détaillées, de la réalité augmentée ou virtuelle à la reconnaissance de la langue des signes. Nous présentons ensuite un état de l'art des approches basées sur la vision par ordinateur, dans le contexte des IHM. L'interprétation visuelle des gestes offre l'interaction la plus naturelle et intuitive, mais elle est aussi la plus difficile à mettre en œuvre.
- ✓ Chapitre 2 : présente les différentes étapes de conception pour la reconnaissance d'un geste de la main nous commençons tout d'abord par la segmentation de la main qui est une étape déterminante pour la suite des traitements. Nous présentons une approche basée sur une modélisation de la couleur de peau. Nous étudierons par la suite la méthode de seuil. Et Nous présentons ensuite l'extraction de caractéristiques et du contour de la main. Et enfin la classification pour la reconnaissance du geste.
- ✓ Chapitre 3 : présente notre système de reconnaissance des gestes de la main, une conclusion générale résume l'importance de notre travail, et enfin nos perspectives envisagées pour son amélioration.

CHAPITRE I

Généralités sur la reconnaissance des gestes de la main

Chapitre 1 Généralités sur la reconnaissance des gestes de la main

I. Introduction

Le geste est un des canaux de communications les plus riches, dans le domaine des Interfaces Homme-Machine (IHM), la main peut servir à pointer (pour remplacer la souris), à manipuler des objets (pour la réalité augmentée ou virtuelle), ou à communiquer par gestes avec un ordinateur.

Par rapport à la richesse de l'information véhiculée par les gestes de la main, les possibilités de communication avec les ordinateurs sont aujourd'hui réduites avec la souris et le clavier. L'interaction homme-machine est basée actuellement sur le paradigme WIMP (Window, Icon, Menu, Pointing device) fenêtre, icône, menu, dispositif de pointage) qui présente les bases fonctionnelles d'une interface graphique informatique.



Figure 1 - WIMP "Fenêtre, Icône, Menu, Pointers" [2]

La majorité des systèmes d'exploitation repose sur ce concept, avec un dispositif de pointage, généralement la souris, qui permet d'interagir avec des éléments graphiques tels que des fenêtres, des icônes et des menus, de façon plus intuitive qu'avec une interface textuelle (en ligne de commande). En utilisant les gestes de la main, l'interface devient perceptuelle.

Les systèmes de reconnaissance de gestes ont d'abord utilisé des gants électroniques munis de capteurs fournissant la position de la main et les angles des articulations des doigts. Mais ces gants sont onéreux et encombrants, d'où l'intérêt croissant pour les méthodes de vision par ordinateur. En effet, avec les progrès techniques et l'apparition de caméras bon marché, il est désormais possible de développer des systèmes de reconnaissance de gestes basés sur la vision par ordinateur, fonctionnant en temps réel.



Figure 2 - Reconnaissance des gestes avec les gants [10]

Toutefois, la main étant un organe complexe, déformable, comportant de nombreux degrés de liberté au niveau des articulations, il est difficile de reconnaître sa forme à partir d'images sans un certain nombre de contraintes.

Avec l'évolution des technologies d'acquisition et des techniques de reconnaissance de gestes, de nombreux domaines d'application ont émergé :

- la reconnaissance de la langue des signes;
- la réalité virtuelle, où la main sert à manipuler des objets virtuels et déclencher des actions, ou à naviguer dans un environnement virtuel;
- la réalité augmentée, où le monde physique est augmenté avec des informations virtuelles, par exemple par une rétroprojection;
- les applications multimodales, associant le geste à d'autres moyens de communication, tels que la parole ou les expressions du visage;
- le codage et la transmission de gestes à bas débit pour la téléconférence;
- la biométrie, pour la reconnaissance de personnes avec la forme de leur main.

1.1. LES GESTES DE LA MAIN

Les gestes sont un des moyens de communication les plus riches que l'être humain possède. Ils permettent d'agir sur le monde physique, et servent aussi à communiquer.

De plus, le geste permet à la fois d'émettre des informations, mais aussi d'en recevoir.

Cadoz [6] définit trois fonctions principales des gestes de la main:

1. *la fonction ergotique*: la main joue le rôle d'organe moteur et agit sur le monde physique pour le transformer. Elle applique aux objets des forces pour les déplacer ou les déformer.
2. *la fonction épistémique* : la main joue le rôle d'organe de perception. Le sens du toucher (sens tactilo-proprio-kinesthésique) donne des informations sur la forme, l'orientation, la distance, la grandeur, le poids, la température, les mouvements des objets, etc.
3. *la fonction sémiotique* : la main joue le rôle d'organe d'expression pour l'émission d'informations visuelles. Cela comprend la langue des signes, le geste Co-verbal, qui accompagne la parole, ou les gestes permettant une communication basique lorsqu'on ne peut pas utiliser la parole, comme dans un environnement bruité ou en plongée sous-marine.

Dans le cadre de la reconnaissance de gestes pour les IHM, nous nous intéressons plus particulièrement à la fonction sémiotique. Cette fonction est la plus riche et la plus complexe. Elle peut être décomposée en plusieurs catégories, et différentes classifications ont été proposées.

1.2. TAXONOMIE DES GESTES DE QUEK

Une classification bien adaptée au domaine de l'interaction homme-machine est la taxonomie de Quek [17], qui décompose les gestes en gestes manipulatifs, correspondant aux fonctions ergotique et épistémique ; et en gestes communicatifs, correspondant à la fonction sémiotique. Les gestes communicatifs sont décomposés en gestes actifs et en gestes symboliques (figure 3).

Les gestes symboliques ne sont pas compréhensibles directement, il faut être initié pour comprendre leur signification. Il s'agit par exemple des gestes des langues des signes. Ils sont décomposés en gestes référentiels, faisant directement référence à un objet ou à un concept (p. ex. le frottement du pouce et de l'index pour évoquer l'argent), et en gestes modélisant, qui modélisent un état ou une opinion et s'emploient souvent en addition à d'autres moyens de communication (p. ex. pour donner une idée de la taille d'un objet). Ces gestes peuvent avoir un sens différent selon la culture.

Les gestes actifs sont directement liés à leur interprétation et sont utilisés en complément de la parole. Ils sont décomposés en gestes mimétiques, consistant à mimer une action, et en gestes déictiques, ou gestes de pointage. Les gestes déictiques sont très utilisés pour l'interaction homme-machine, car le doigt représente un dispositif de pointage naturel et très pratique [2].

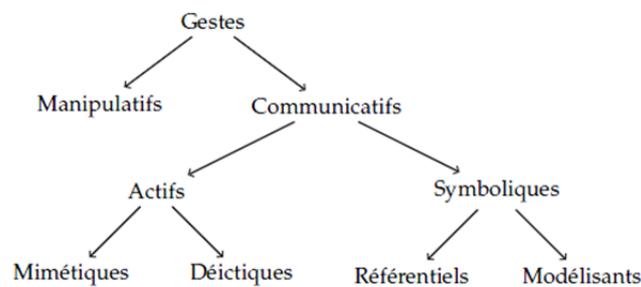


Figure 3 - Taxonomie des gestes de Quek [2]

1.3. GESTES STATIQUES ET DYNAMIQUES

Il existe deux autres catégories de gestes : les gestes statiques, ou postures, et les gestes dynamiques (figure 4). Par ailleurs, il faut distinguer la position et la configuration de la main. En combinant ces deux aspects, on obtient la classification proposée par Harling et Edwards [8]:

- position statique, configuration statique (les postures) ;
- position statique, configuration dynamique ;
- position dynamique, configuration statique (p. ex. les gestes de pointage) ;
- position dynamique, configuration dynamique (p. ex. la langue des signes).

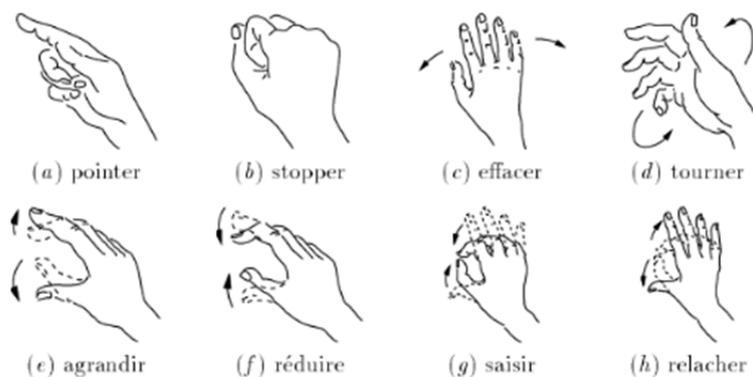


Figure 4 - "Exemple de gestes statiques & Dynamique"[5]

II. Etat de l'art « Interaction Homme – Machine »

Cette section présente les différents dispositifs permettant d'interagir avec un ordinateur. Nous présentons ensuite les applications et les nouvelles possibilités d'interaction basées sur la reconnaissance des gestes de la main.

2.1. DISPOSITIFS D'INTERACTION

La majorité des systèmes d'exploitation repose sur le paradigme **WIMP** (Window, Icon, Menu, Pointing device) [2], avec un dispositif de pointage, généralement la souris, qui permet d'interagir avec des éléments graphiques tels que des fenêtres, des icônes et des menus. Il existe aussi des interfaces haptiques, qui permettent un retour d'information à l'utilisateur, avec un retour de toucher ou un retour d'effort. La perception tactilo kinesthésique, ou haptique, résulte de la stimulation de la peau par le contact avec des objets.

Cette section présente les différents dispositifs permettant une interaction avec l'ordinateur : les périphériques d'entrée (souris et autres dispositifs de pointage), les gants de données, les caméras vidéo et les écrans tactiles.

2.1.1. PÉRIPHÉRIQUES D'ENTRÉE

Afin d'interagir avec un ordinateur, la souris s'est imposée comme le périphérique d'entrée indispensable [2]. Il existe aussi la boule de commande, ou « trackball » figure 5(a). Le pavé tactile ou « touchpad » est un dispositif de pointage spécifique aux ordinateurs portables, permettant de remplacer la souris. Il s'agit d'une surface sensible de faible dimension, utilisant la capacité électrique.

Pour les jeux vidéo, le périphérique usuel est la manette ou « joystick ». Récemment, les constructeurs ont cherché à développer d'autres types d'interactions que les boutons, avec par exemple des vibrations pour le retour d'effort, ou des capteurs de mouvements. Ainsi, la Wiimote 1 figure 5(b) de Nintendo a marqué un tournant dans les périphériques de jeux vidéo. Cette manette est équipée de capteurs qui lui permettent de se repérer dans l'espace et de retranscrire les mouvements de l'utilisateur à l'écran.



Figure 5 - (a) Trackball – (b) la Wiimote de Nintendo

2.1.2. LES GANTS DE DONNÉES

Les gants de données (ou gant électronique, gant numérique), qui sont munis de capteurs fournissant la position de la main et les angles des articulations des doigts.

Les gants de données sont utilisés de longue date pour la reconnaissance de la langue des signes [5], car ils fournissent les positions précises et fiables des articulations de la main. Malheureusement, ces gants ont un coût élevé et sont encombrants, leur utilisation est contraignante pour l'utilisateur.



Figure 6 - Les Gants de données [10]

2.1.3. ÉCRANS TACTILES

Les écrans tactiles combinent à la fois entrée et sortie, avec la visualisation sur un écran et un dispositif de pointage pour interagir directement avec l'information affichée à l'écran. Cette technologie est utilisée pour des écrans de la taille d'un moniteur d'ordinateur, par exemple pour les guichets de billetterie automatique ou pour les « assistants personnels digitaux ». Il existe différentes technologies pour les écrans tactiles : capacitive, résistive, infrarouge, à ondes de surface... De manière générale, les écrans tactiles souffrent de différents inconvénients : ils sont onéreux, limités en surface, sujet à l'usure et sensibles aux rayures (suivant la technologie employée).

Les écrans tactiles permettent la reconnaissance de gestes simples, appelés gestes de dessins. Par exemple, Mertz et al [7] utilisent les gestes de commande présentés sur la figure 7.a. Il existe aussi des alphabets simplifiés pour les PDA (figure 7.b), permettant la réalisation de lettres d'un seul trait.

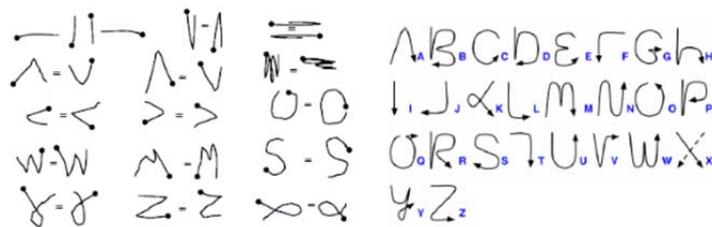


Figure 7 - (a) Geste de commande – (b) Alphabets simplifié [7]

2.1.4. LES CAMERAS VIDEO

Contrairement aux systèmes précédents, les caméras vidéo captent les mouvements de la main sans que l'utilisateur ne soit contraint de porter un équipement particulier, ou d'utiliser un périphérique dédié. Toutefois, pour certains systèmes, des marqueurs ou un gant coloré sont utilisés pour faciliter la détection des différentes parties de la main. La difficulté de cette approche est de mettre au point des traitements robustes pour interpréter le flux vidéo et extraire l'information utile de la grande quantité d'information disponible. Par ailleurs, une caméra ne fournit qu'une information 2d. Pour obtenir des informations en 3d, il faut utiliser deux ou plusieurs caméras, ou une modélisation 3d. Par conséquent, les occultations sont un problème important, inhérent à la projection de l'espace 3d dans une image. Avec l'augmentation de la puissance des ordinateurs de bureau et l'apparition de caméras bon marché, il est désormais possible de développer des systèmes de reconnaissance de gestes fonctionnant en temps réel. C'est à cette problématique que nous nous sommes intéressés dans le cadre de ce PFE. Les caractéristiques telles que le taux de rafraîchissement ou la résolution varient d'une caméra à l'autre. Des valeurs élevées de ces caractéristiques sont avantageuses pour avoir une mise à jour fréquente des images et un niveau de détail important. Toutefois, un compromis est nécessaire car des valeurs trop importantes augmentent la complexité et le temps de traitement des données [2].

Par ailleurs, les caractéristiques du capteur (CCD ou CMOS) et de l'optique ont une influence primordiale sur la qualité de l'image obtenue et la sensibilité à l'éclairage.

2.2. APPLICATION ET NOUVELLES POSSIBILITES D'INTERACTION

Cette section présente différentes applications, et notamment les surfaces interactives, ainsi que quelques applications récentes destinées au grand public.

2.2.1. RECONNAISSANCE DE LA LANGUE DES SIGNES

La reconnaissance de la langue des signes est une application naturelle de la reconnaissance de gestes. En effet, un système de traduction automatique des signes est particulièrement intéressant pour les sourds et muets. Braffort [5] présente une étude détaillée de la (LSF) Langue des Signes Française. Cette langue possède un vocabulaire et une syntaxe, et repose sur de nombreux paramètres :

- la configuration : pour représenter les objets ;
- le mouvement : pour représenter les actions ;
- l'emplacement : pour indiquer où sont effectuées les actions ;
- l'orientation : pour conjuguer certains verbes, ou préciser l'orientation des objets ;
- la mimique faciale : pour exprimer le mode du discours.

Ce bref aperçu révèle la richesse et la complexité de la LSF (Langue des Signes Française), et il en est de même pour les autres langues. C'est pourquoi la grande majorité des systèmes de reconnaissance de la langue des signes utilise des gants numériques, qui permettent d'obtenir des paramètres sur la configuration de la main plus facilement qu'avec la vision.

Dans les travaux consacrés à la reconnaissance de la langue des signes en vision par ordinateur (Ong et Ranganath [12], le vocabulaire est généralement restreint à un sous-ensemble de gestes. La référence dans ce domaine est les travaux de Starner et Pentland [13] [14], qui s'intéressent à la reconnaissance de la langue des signes américaine (ASL American Sign Language), avec un vocabulaire de 40 mots. Ils obtiennent un excellent taux de reconnaissance de plus de 90%.

2.2.2. REALITE VIRTUELLE

La réalité virtuelle consiste à plonger l'utilisateur dans un environnement d'images de synthèse. Cette immersion dans un environnement virtuel peut se faire grâce à un casque, ou dans une pièce dédiée munie de plusieurs écrans ou d'un vidéoprojecteur [9] [15].

L'utilisateur est donc en immersion complète dans un environnement dans lequel il peut interagir, notamment par le biais de gestes.



Figure 8 - Réalité Virtuelle

2.2.3. REALITE AUGMENTEE

La réalité augmentée mélange le monde physique et des informations virtuelles, en superposant ces informations dans le champ de vision de l'utilisateur [2].

Ce type de système est fondé sur une analyse par l'ordinateur du monde environnant l'utilisateur, au moyen d'un système de vision, de type caméra vidéo par exemple, ou de capteurs spécifiques. Des informations virtuelles sont alors projetées sur des éléments réels, par exemple par l'intermédiaire d'un vidéo projecteur.

Une des applications principales de la reconnaissance de gestes en réalité augmentée est de rendre une surface interactive : une table, un tableau ou un bureau, sur laquelle des images sont projetées. L'utilisateur peut alors interagir avec des objets réels ou virtuels.



Figure 9 - Réalité augmentée

2.2.4. CONTROLE MACHINE :

L'interaction gestuelle joue un rôle important pour le contrôle des machines, ce qui s'explique par son efficacité pour la commande. Plusieurs travaux ont été développés dans ce contexte, parmi eux un travail sur l'exploitation du geste de la main pour remplacer la souris afin de transformer un écran classique en un écran tactile et de permettre à l'utilisateur de se servir de son doigt comme dispositif de pointage.

Dans le but d'établir une communication naturelle entre l'homme et un robot personnel, fut développée une interaction basée sur le geste de la main (geste de pointage), et la parole pour informer un robot mobile, qui est doté de capteur sonore et d'un système de stéréovision, sur la localisation d'un objet dans son environnement



Figure 10 - Souris gestuelle

III. Les Difficultés liées à la vision par ordinateur

Contrairement aux gants numériques, qui permettent d'obtenir directement des informations sur la position et la configuration de la main [2], la conception d'un système de vision par ordinateur nécessite la prise en compte d'un certain nombre de difficultés :

1. Eclairage de la scène: les variations de l'éclairage de la scène ont un impact important sur les algorithmes de segmentation et d'extraction des caractéristiques, en provoquant des variations de luminosité dans l'image.
2. Ombres : la main projette une ombre dans la scène, même si la luminosité est contrôlée. Les ombres sont parfois segmentées avec la main, suivant la méthode utilisée, ce qui rend le résultat de segmentation difficilement exploitable.
3. Objets de la scène : la scène peut comporter des objets mobiles tels que le clavier, la souris ou des documents. L'algorithme de segmentation de la main doit pouvoir s'adapter à des changements du fond de la scène et distinguer la main des autres objets.
4. Camouflage: lorsque les caractéristiques de l'objet sont proches de celles du fond, celui-ci est difficilement détectable. Dans le cas de la main, ce problème se produit par exemple lorsque le bureau a une couleur de teinte chair.
5. Occultations : les occultations se produisent lorsqu'un objet est masqué par un autre, ce qui peut arriver lorsqu'on utilise les deux mains. Dans le cas de la main, il existe un problème supplémentaire avec les « Auto occultations »: la main étant un objet complexe et déformable, il arrive fréquemment que certaines parties, telles que les doigts, soient cachées par d'autres.
6. Disposition des caméras : la distance entre la caméra et les objets joue un rôle important, de même que l'angle de vue des caméras (90° dans notre cas) qui peut causer des déformations géométriques dans l'image.

Pour résoudre ces difficultés, différentes hypothèses sont émises en fonction de l'application visée, sur l'éclairage, sur le type d'objets présents dans la scène ou sur le type de gestes à reconnaître. Par exemple, pour segmenter la main de façon robuste aux variations de luminosité.

IV. Conclusion

Dans ce chapitre, nous avons cité l'utilisation d'un nouveau moyen de communication dans les interactions Homme-Machine qui est « le geste », et plus précisément le geste de la main qui représente une communication naturelle et l'une des méthodes les plus utilisées uniquement chez les personnes qui ont une certaine difficulté à parler ou à entendre. Ou dans certains environnements, à titre d'exemple l'environnement sous-marin, où il est inconfortable d'utiliser des appareils vocaux ou des dispositifs d'entrée typique.

En dernier nous avons cité les problèmes rencontrés pour le développement de notre système de reconnaissance de geste et afin de réussir notre système, une bonne configuration et un choix judicieux des méthodes s'imposent.

CHAPITRE II

Système de Reconnaissance des gestes de la main

Chapitre 2 Système de Reconnaissance des gestes de la main

I. Introduction

Le processus de reconnaissance de formes, est proche du raisonnement de l'être humain, d'où s'inspirent les domaines de l'intelligence artificielle, qui servent à automatiser des tâches humaines. Notre système de reconnaissance de geste est schématisé dans la figure ci-dessous, résumant les étapes qui seront détaillées par la suite pour l'identification du geste.

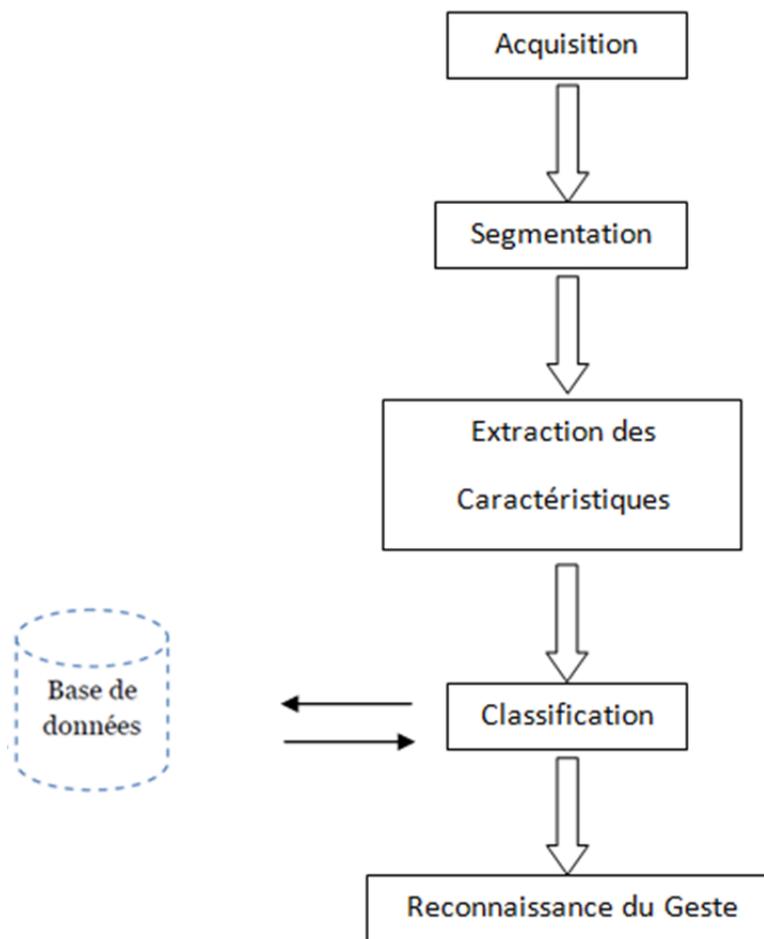


Figure 11 - Schéma général d'un système de reconnaissance des gestes

II. Système de reconnaissance des gestes de la main

La reconnaissance de gestes à l'aide de la webcam est une option attrayante pour remplacer l'interaction homme-machine à l'aide d'une souris.

De nos jours, trois méthodes sont utilisées principalement pour effectuer la reconnaissance gestuelle.

2.1. TABLETTES GRAPHIQUES & ECRANS TACTILE

Ces techniques sont destinées pour la reconnaissance de l'écriture manuscrite, le geste de dessins et les gestes de pointage. Ces techniques ont pour objectif de remplacer les traditionnels menus. Le crayon et la tablette graphiques sont d'usage courant en Conception Assistée par Ordinateur (CAO) [7].



Figure 12 - Ecran Tactile & Tablette graphique

2.2. PORTABLE DISPOSITIFS ELECTROMAGNETIQUES, GANTS NUMERIQUE :

Les gants spéciaux principalement utilisé dans l'industrie du cinéma. Il fonctionne bien mais est coûteux [10].



Figure 13 - Gant Numérique

2.3. LA VISION PAR ORDINATEUR :

La vision par ordinateur (aussi appelée vision artificielle ou vision numérique) est une branche de l'intelligence artificielle dont le principal but est de permettre à une machine d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition (dans notre système nous allons opter pour une Webcam) [3].

La plupart des systèmes de reconnaissance du geste utilisent cette technique, car son aspect intuitif et simple reste un moyen plus approprié pour l'interaction Homme-Machine.

2.3.1. Qu'est ce que la vision

Le monde a une structure 3D et est composé d'objets, l'être humain sait parfaitement décrire et interpréter ce monde, pourtant, l'information disponible sur la rétine n'est qu'un ensemble de points (environ un million de pixels). Chaque pixel contient des informations sur la lumière (quantité et contenu spectral/couleur) reçue en ce point de la rétine, les objets (téléphone, voiture...) n'existent pas sur la rétine, et pourtant on les voit : leur interprétation est le résultat du processus visuel [3].

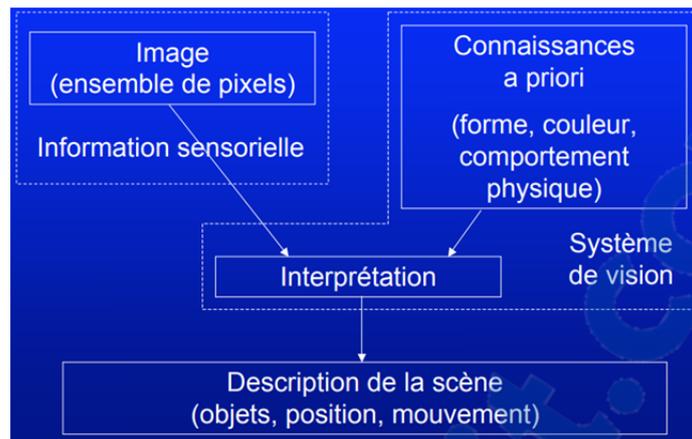


Figure 14 - Théorie de la vision [3]

2.3.2. Pourquoi la vision

L'utilisation des capteurs de vision est particulièrement intéressante en raison de la grande richesse des informations qu'une caméra peut fournir et la variété de tâches qu'elle permet de réaliser [4].

Un système de vision comporte en général :

- Une source lumineuse pour éclairer la scène.
- Un capteur.
- Une interface (carte d'acquisition) entre le capteur et l'ordinateur qui convertit l'information analogique en information numérique.
- Un système de traitement (ordinateur et logiciels).

2.3.3. Capteurs d'acquisition :

Le capteur est destiné à recevoir les signaux lumineux provenant de la scène à analyser. Il comprend une surface sensible recueillant les informations lumineuses à travers un organe de focalisation (objectif) [4].

Dans notre travail, nous avons utilisé la Webcam qui présente les avantages suivants :

Prix abordable.

- Facilité d'utilisation.
- Mobilité.

Faible résolution (qui est dans notre cas nécessaire pour réduire le temps de transmission).

III. La segmentation

Cette première étape est primordiale, Il existe de nombreuses méthodes, plus ou moins performantes suivant les suppositions faites sur la scène, et suivant l'environnement (intérieur, extérieur, condition d'illumination...) [2].

De plus, nous souhaitons non seulement détecter la main (c.-à-d. savoir si elle est présente dans l'image), mais aussi la segmenter (c.-à-d. savoir quels pixels de l'image appartiennent à la main) afin d'obtenir son contour, avec la meilleure précision possible, pour en extraire des informations sur sa forme et sa position.

Par ailleurs, il faut prendre en compte la contrainte temps réel, car la segmentation de la main n'est que la première étape d'un système de reconnaissance de gestes, qui en compte plusieurs autres. Il faut donc trouver un compromis entre les performances de la segmentation et la rapidité d'exécution.

3.1. LA SEGMENTATION PAR DETECTION DE LA COULEUR DE PEAU

L'analyse de la couleur de la peau est très utilisée pour la détection du visage et des mains. Il existe de nombreux espaces colorimétriques, les plus utilisés étant RGB, HSV et YCbCr. Les performances de ces espaces on démontrait que les résultats sont très similaires, quelque soit l'espace couleur.

3.1.1. RGB

C'est une approche très utilisée dans le domaine de la localisation des visages et de mains, elle repose sur la distinction entre les pixels qui correspondent à la couleur de la peau de celle appartenant au fond de la scène [4]. Il existe plusieurs seuils dans différents espace colorimétrique (RGB, HSV, YCbCr ...), pour notre segmentation, nous avons travaillé dans l'espace colorimétrique RGB.

Le pixel (R, G, B) est classifié comme pixel de peau si :

- **Cas éclairage fort:**

$$(R > 95) \text{ et } (G > 40) \text{ et } (B > 20) \text{ et } ((\max(R, G, B) - \min([R, G, B])) > 15) \text{ et } (ABS(R - G) > 15) \text{ et } (R > G) \text{ et } (R > B)$$

(1)

- **Cas éclairage faible :**

$$(R > b) \text{ et } (G > B) \text{ ou } ((R > 220) \text{ et } (G > 210) \text{ et } (B > 170) \text{ et } (ABS(R - G) \leq 15))$$

(2)

Algorithme de segmentation par la couleur de peau dans l'espace RBG éclairage fort:

Si $(R > 95) \ \& \ (G > 40) \ \& \ (B > 20) \ \& \ (\max(R, G, B) - \min(R, G, B)) > 15 \ \& \ \text{abs}(R, G, B) \ \& \ (R > G) \ \& \ (R > B);$

Alors $(\text{Image}(R, G, B)) = 0$;

Sinon niveau de gris $(\text{Image}(R, G, B)) = 255$;

L'avantage que porte cette méthode est que la caméra peut être en mouvement, mais l'inconvénient est que cette dernière est sensible aux conditions d'éclairage et la présence d'éléments ayant des couleurs proches de la couleur de la peau.

3.1.2. YCbCr chrominance de luminance

Cet espace de couleur est utilisé dans les informations de couleur vidéo numérique représentent deux couleurs Cb et Cr, Cb est différent entre bleu et Cr est la différence entre les références de la composante rouge de la valeur [1].

Ceci est essentiellement la transformation RGB en YCbCr pour la séparation de la luminance et de la chrominance.

3.1.3. HSV Teinte, Saturation et valeur

En HSV, la teinte détermine la couleur dominante et la saturation définissent la couleur, tandis que la valeur mesure l'intensité ou la luminosité [1]. C'est assez bien pour choisir une seule couleur mais elle ignore la complexité de l'apparence de la couleur. Ils échangent la vitesse de calcul signifie la pertinence coûteuse et perceptive de calcul

IV. Seuillage (thresholding)

L'étape la plus importante pour la segmentation manuelle est le seuillage, qui est utilisé pour séparer la main de l'arrière-plan. Le seuillage peut être utilisé comme méthode d'extraction de l'arrière-plan en affectant certaines valeurs pour chaque pixel comme valeur de seuil. Le seuillage produira une image binaire avec tous les pixels de 1 représentant la main. Cela se produit lorsque le pixel dont l'intensité est inférieure à la valeur seuil est défini sur 0, tandis que tout pixel dont l'intensité est supérieure à la valeur seuil est défini sur 1.

Le seuillage d'une image en niveaux de gris I consiste à associer la valeur 0 (noir) à tous les pixels dont la valeur est inférieure au seuil α et la valeur 1 (blanc) à tous les autres. On obtient une image seuillée S , aussi appelée masque binaire ou silhouette :

$$\forall(x, y), S(x, y) = \begin{cases} 1, & \text{si } I(x, y) \geq \alpha \\ 0, & \text{sinon} \end{cases}$$

L'inconvénient de cette méthode est qu'il faut déterminer le seuil manuellement, de façon empirique, et qu'un seuil donné est adapté à des conditions d'illumination bien précises. Or, il est très difficile de trouver un seuil qui donne de bons résultats dans toutes les situations.

Il existe des méthodes plus robustes où le seuil est calculé automatiquement en fonction de l'image. C'est le cas de la méthode de Otsu [16], pour laquelle le seuil est calculé à partir de l'histogramme des niveaux de gris de l'image.

Cette méthode est adaptée à des images dans lesquelles les deux classes sont bien définies. L'histogramme $h(i)$ de l'image est séparé en deux classes, dont on calcule les moyennes μ_1 et μ_2 , et les variances σ_1^2 et σ_2^2 , en fonction du seuil T :

$$\begin{aligned} \mu_1 &= \frac{1}{T} \sum_{i=0}^{T-1} h(i) & \sigma_1^2 &= \frac{1}{T} \sum_{i=0}^{T-1} (h(i) - \mu_1)^2 \\ \mu_2 &= \frac{1}{256 - T} \sum_{i=T}^{255} h(i) & \sigma_2^2 &= \frac{1}{256 - T} \sum_{i=T}^{255} (h(i) - \mu_2)^2 \end{aligned}$$

Si l'image comporte N pixels, on a les probabilités des deux classes :

$$p_1 = \frac{1}{N} \sum_{i=0}^{T-1} h(i) \quad p_2 = \frac{1}{N} \sum_{i=T}^{255} h(i) = 1 - p_1$$

Le seuil optimal est celui qui minimise la variance intra-classe σ_w^2 :

$$\sigma_w^2 = p_1 \sigma_1^2 + p_2 \sigma_2^2$$

La figure 15 montre le résultat du seuillage avec la méthode d'Otsu et avec un seuil fixe.

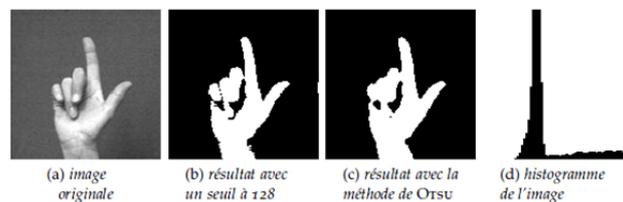


Figure 15 - Exemple de Segmentation par seuillage

v. Contours

Une fois obtenue la masse uniforme correspondant à l'extraction de la main (segmentation), nous allons majoritairement travailler sur ses contours. Ceux-ci sont extraits avec un simple filtre gradient, puis lissés.

Les contours peuvent être utilisés de plusieurs manières, pour l'extraction de caractéristiques, et ce pour les paramètres qu'on puisse calculer à partir de ceux-ci.

5.1. FILTRES DE DETECTION DE CONTOURS :

Nous avons deux approches [11] :

- Approche gradient : détermination des extréma locaux dans la direction du gradient.
- Approche laplacien : détermination des passages par zéro du laplacien.

5.1.1. Approche gradient :

Cette approche consiste à approximer les dérivées par différence :

$$\nabla_u I(u, v) = I(u, v) - I(u - n, v),$$

ou :

$$\nabla_u I(u, v) = I(u + n, v) - I(u - n, v),$$

avec, en général $n = 1$.

Ces dérivées sont calculées par convolution de l'image avec un masque de différences.

i. Opérateurs de Roberts (1962)

$$h1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\phi = \arctan(I * h2 / I * h1) + \pi/4.$$

- Forte sensibilité au bruit en raison de la taille des masques

ii. Opérateurs de Prewitt

$$h1 = 1/3 \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad h2 = 1/3 \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Masques de calcul du gradient en x et y.

La convolution de l'image par les masques ci-dessus correspond aux calculs des dérivées de l'image pré-filtrée par un filtre h séparable :

$$h(x, y) = h(x) h(y),$$

avec : $h = \frac{1}{3}[1 \ 1 \ 1]$ et $d = \nabla h = [-1 \ 0 \ 1]$. En effet :

$$h1(x, y) = d(x) s(y),$$

$$h2(x, y) = s(x) d(y),$$

- Moins sensible que le filtre de Roberts

iii. Opérateurs de Sobel 1972

$$h1 = 1/4 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h2 = 1/4 \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Très populaire
- Meilleure estimation que celle de Prewitt.

5.1.2. Approches Laplacien :

Les points de contour sont caractérisés par des passages par zéro du laplacien.

La détection de ces points s'effectue en deux étapes :

1. Détection des passages par zéros. Les pixels pour lesquels laplacien change de signe sont Sélectionnés.
2. Seuillage des passages par zéros de fortes amplitudes (par hystérésis par exemple).

La méthode la plus simple est la convolution avec un des masques suivants :

$$M_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1-8 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Ou} \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1-4 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

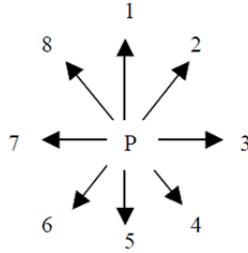
- Le gradient est caractérisé par sa sensibilité accrue au bruit par rapport au Laplacien.

VI. Codage

Le problème est ici de savoir comment représenter les informations graphiques de la main segmentée, de manière à détecter des caractéristiques.

6.1. CODAGE DE FREEMAN

Nous travaillons maintenant sur les contours de la main, qui vont être parcourus, et au fur et à mesure de cet avancement on note l'évolution entre chaque point de contour. Ainsi, pour chaque point, on admet 8 directions possibles afin d'arriver au point suivant [4]:



Les points déjà visités doivent être marqués pour ne pas « boucler », et l'ordre de test privilégie d'abord les angles à 45° (directions 2,4,6,8) puis les directions parallèles au repère image (directions 1,3,5,7).

Le contour est un contour fermé, que l'on va rendre cyclique. C'est à dire que le point qui suit le « dernier » point de contour nous ramène au point « initial » (au niveau du parcours du tableau codant ce contour). Le point « initial », d'où a été lancé l'appel à la fonction de suivi, est le point de contour de plus faible ordonnée (le plus « haut » dans l'image segmentée, c'est à dire le sommet du majeur dans la posture d'initialisation du système.

VII. Classification

Une approche heuristique simple de la classification est un ensemble de règles explicites IF-THEN qui se réfèrent à la caractéristiques de la cible et exigent qu'ils se situent dans une certaine gamme qui est typique d'un spécifique geste [23].

Les classifications sont également utilisées comme une étape primaire pour les algorithmes de classification des gestes dynamiques.

IF a < $\Theta a N 2$ **THEN the observed gesture is the stop**
IF a >= $\Theta a N 2$ **THEN the object is the hand.**

Le seuil spécifié pour c a été déterminé expérimentalement à partir d'un ensemble de données de plusieurs productions du geste et de tous les autres gestes, réalisées par différents utilisateurs.

VIII. Conclusion

Dans ce chapitre nous avons étudié, les différentes étapes d'un système de reconnaissance de geste. Dans notre système, nous avons opté pour l'approche par vision, pour cela le traitement d'image est nécessaire, ce qui permet d'extraire les caractéristiques utilisées pour reconnaître le geste. Le choix de la méthode est défini par la pertinence de l'information qu'elle met en évidence par rapport au problème posé.

Dans le chapitre suivant, nous allons détailler les étapes et justifier le choix des différentes méthodes utilisées.

CHAPITRE III

Système de Reconnaissance des gestes de la main

Chapitre 3 Implémentation & Résultats

I. Introduction

Dans notre projet de fin d'études qui consiste à l'implémentation d'un système de reconnaissance des gestes de la main qui a pour objectif d'améliorer l'interaction homme machine, ce dernier chapitre porte notre choix sur les différentes méthodes utilisées pour la réalisation de ce système. Ci après une représentation générale de notre système de reconnaissance des gestes de la main

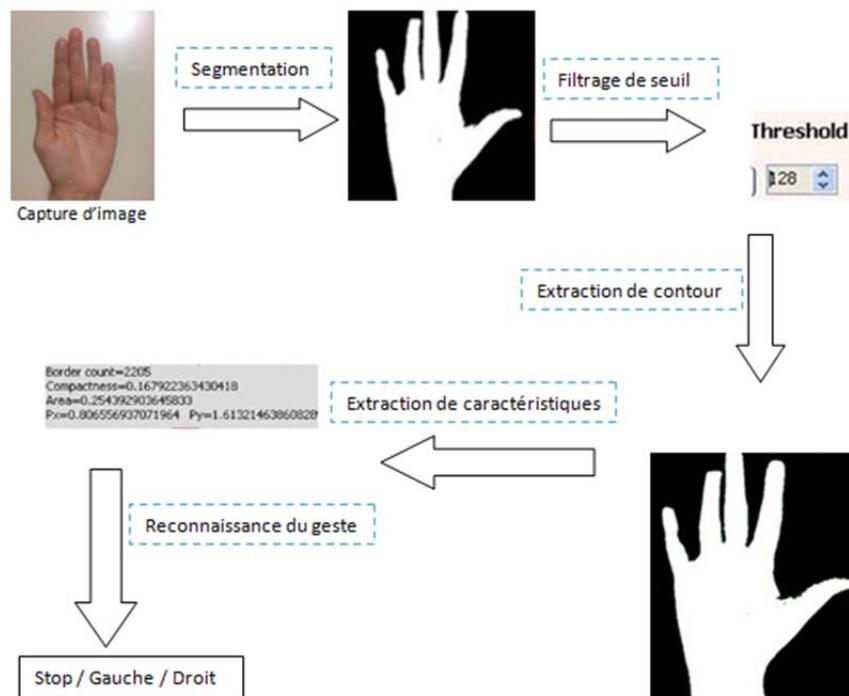


Figure 16 - Etapes de l'Algorithme de reconnaissance

II. Environnement de développement

Nous avons opté pour le langage de programmation c # qui est un langage de programmation orienté objet qui est défini par les caractéristiques suivantes :

- **Récent** : il a été créé par Microsoft au début des années 2000, ce qui en fait un langage assez jeune (comparé au C, ou C++ et Java).
- **Libre** : le langage est ouvert, et on peut s'en servir pour développer des applications sous Linux notamment.
- **Inspiré de Java** : même s'il a ses spécificités, il ressemble globalement plus au Java qu'au C ou au C++ contrairement à ce que son nom pourrait laisser penser.
- **Associé au framework .NET** : un langage seul comme le C# ne permet pas de faire grand chose. On l'associe en général à une boîte à outils que l'on appelle le framework .NET (aussi écrit Dot NET) qui offre toutes les possibilités que l'on peut imaginer : accès réseau, création de fenêtres, appel à une base de données...

- **Apprécié en entreprise** : si Java reste un des langages les plus utilisés en entreprise, C# se présente comme un sérieux challenger. C'est aujourd'hui une compétence recherchée en entreprise.

III. Architecture matériel

Nous avons développé notre projet sur un PC ayant les caractéristiques suivantes :

- Processeur : intel core i5 2.27 GHZ
- RAM : 4Go
- Systeme d' exploitation : Windows 7 – 64 bits
- WebCam: intégrée

IV. Système de reconnaissance du geste

Notre application de reconnaissance des gestes de la main capture des images à partir d'une Webcam en temps réel.

L'utilisation d'une image Webcam est capturée, traitée pour la détection de la main, des Caractéristiques, les gestes sont identifiés, il y a 3 types de gestes qui peuvent être reconnus: stop, droite et gauche.

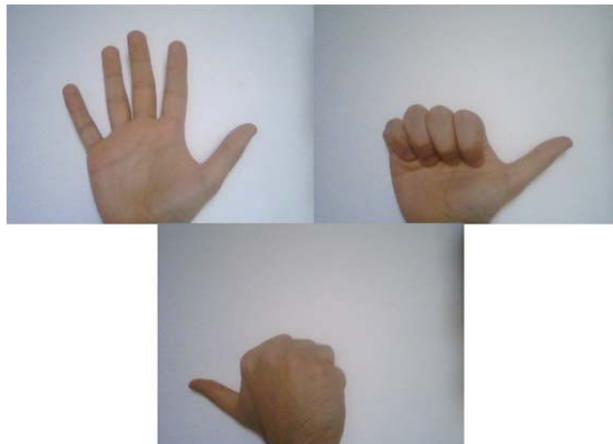


Figure 17 - Les Trois types de gestes

Les grandes étapes de notre système est comme suit :

4.1. ACQUISITION

Parmi les différentes méthodes d'acquisition que nous avons cité dans le chapitre 2, notre choix s'est porté sur l'approche par vision, et pour cela nous avons traité notre capture d'image en utilisant AForge.Net Framework [22] qui est une bibliothèque de vision artificielle et d'intelligence artificielle développée à l'origine par Andrew Kirillov pour le .NET Framework.

4.2. SEGMENTATION

4.2.1. RGB :

RGB est un pixel d'espace couleur de 3D où chaque pixel a une combinaison de trois couleurs, rouge, vert et bleu à un emplacement spécifique [20]. Dans la reconnaissance des gestes, la couleur est la caractéristique la plus fréquemment utilisée pour la localisation des mains puisque la forme et la taille de la main la projection dans le plan d'image bidimensionnel varie grandement. Un discret en trois dimensions histogramme $h_{object}(r, g, b)$ peut être utilisé pour représenter les dimensions correspondant au rouge, composants verts et bleus. La somme totale de h_{object} sur toutes les couleurs est donc égale à nombre de pixels d'objets considérés n_{object} ,

$$\sum_r \sum_g \sum_b h_{object}(r, g, b) = n_{object} \quad (1)$$

Étant donné un pixel de l'objet, la probabilité qu'il ait une certaine couleur (r, g, b) peut être calculé à partir de h_{object} comme

$$P(r, g, b | object) = h_{object}(r, g, b) / n_{object} \quad (2)$$

En créant un histogramme complémentaire h_{bg} des couleurs de fond, nous aurons la probabilité pour l'arrière-plan de la même manière:

$$P(r, g, b | object) = \frac{h_{bg}(r, g, b)}{n_{bg}} \quad (3)$$

En appliquant la règle de Bayes, la probabilité de tout pixel représentant une partie de l'objet peut être calculé à partir de sa couleur (r, g, b) en utilisant les équations (2) et (3)

$$P(object | r, g, b) = \frac{P(r, g, b | object) P(object)}{P(r, g, b | object) P(object) + P(r, g, b | bg) P(bg)} \quad (4)$$

$P(object)$ et $P(bg)$ désignent les probabilités a priori d'objet et de fond, respectivement, avec $P(object) + P(bg) = 1$.

L'image de probabilité d'objet « $I_{obj, prob}$ », $prob$ est créée à partir de

$$I_{obj, prob}(x, y) = P(object | I(x, y)) \quad (5)$$

Une structure de données appropriée pour représenter cette classification est un masque binaire

$$I_{obj, prob}(x, y) = \begin{cases} 1 & \text{if } I_{obj, prob}(x, y) \geq \theta (target) \\ 0 & \text{otherwise (background)} \end{cases} \quad (6)$$

4.2.2. Méthode de seuillage :

Le calcul automatique du seuil [18] de probabilité d'objet Θ sans l'utilisation de connaissances de haut niveau est la suivante :

1. Définir arbitrairement un ensemble de pixels de fond (certains ont généralement un a priori Elevé probabilité de fond, par ex. les quatre coins de l'image). Tous les autres pixels sont définis comme premier plan. Ceci constitue une classification initiale.
2. Calculer les valeurs moyennes pour le fond et le premier plan, μ_{objet} et μ_{bg} , en fonction de la classification la plus récente. Si les valeurs moyennes sont identiques à celles calculées dans l'itération précédente, arrêtez.
3. Calculer un nouveau seuil $\Theta = 1/2 (\mu_{\text{objet}} + \mu_{\text{bg}})$ et effectuer une autre classification de tous pixels, puis passez à l'étape 2.

NB : Filtrage de seuil de l'image binaire en utilisant une valeur supérieure et inférieure
Dans notre application l'utilisation de ce dernier ce fait de façon manuelle

4.3. EXTRACTION DE CONTOUR ET DES CARACTERISTIQUES

4.3.1. DESCRIPTION DE LA REGION DE LA MAIN

L'algorithme pour trouver les points de bordure de toutes les régions d'une image est le suivant [19] [21]:

- 1- Créer une matrice d'aide m avec les mêmes dimensions que $lobj$, masquer et initialiser toute les entrées à 0. Définissez (x, y) les coordonnées actuelles et initialisez-les à $(0,0)$. Définir (x', y') et (x'', y'') comme des coordonnées temporaires.
- 2- Itérer de gauche à droite dans toutes les rangées d'images successivement, en commençant par $y = 0$.
- 3- Créer une liste B de points de bordure et stocker (x, y) comme premier élément.
- 4- Set $(x', y') = (x, y)$
- 5- Scanner le voisinage 8 de (x', y') en utilisant (x'', y'') , en commençant par le pixel qui suit le dernier pixel stocké dans B dans le sens inverse des aiguilles d'une montre, ou à $(x'-1, y'-1)$ si B contient seulement un pixel. Continuez dans le sens antihoraire en sautant les coordonnées qui se trouvent en dehors de l'image, jusqu'à $lobj$, masque $(x'', y'') = 1$. Si (x'', y'') est identique au premier élément de B , aller à l'étape 6. Else store (x'', y'') dans B . Set $(x', y') = (x'', y'')$ et goto étape 5.
- 6- Itérer à travers B , en considérant, pour chaque élément est le successeur de la dernière, qui est le prédécesseur du premier. Si $y_{i-1} = y_i + 1 \neq y_i$, mettez $m(x, y) = 1$ pour indiquer que la bordure touche la ligne $y = y_i$ à x_i . Sinon, si $y_{i-1} \neq y_i \vee y_i \neq y_i + 1$, mettez $m(x_i, y_i) = 2$ pour indiquer que la frontière intersecte avec la ligne $y = y_i$ à x_i .
- 7- Ajoutez B à la liste des bordures calculées et passez à l'étape 2.

4.3.2. LES CARACTERISTIQUES GEOMETRIQUES

Les principales caractéristiques géométriques [21] à extraire sont les suivantes

- La longueur de la bordure est 1 si leurs coordonnées x ou y sont égales, et $\sqrt{2}$ autrement dit elle dépend de l'échelle / résolution, et est invariant de translation et de rotation.
- Zone, centre de gravité (cog) et moments de second ordre

Dans un algorithme efficace pour le calcul des moments arbitraires $\nu_{p,q}$ de ν polygonis présenté. La zone $a = \nu_{0,0}$, ainsi que les moments normalisés. Depuis ces équations exigent que le polygone soit fermé

$$\nu_{0,0} = a = \frac{1}{2} \sum_{i=1}^n (x_{i-1} y_i - x_i y_{i-1}) \quad (7)$$

$$\alpha_{1,0} = x_{cog} = \frac{1}{6a} \sum_{i=1}^n (x_{i-1} y_i - x_i y_{i-1}) (x_{i-1} + x_i) \quad (8)$$

$$\alpha_{0,1} = y_{cog} = \frac{1}{6a} \sum_{i=1}^n (x_{i-1} y_i - x_i y_{i-1}) (y_{i-1} + y_i) \quad (9)$$

$$\alpha_{2,0} = \frac{1}{12a} \sum_{i=1}^n (x_{i-1} y_i - x_i y_{i-1}) (x_{i-1}^2 + x_{i-1} x_i + x_i^2) \quad (10)$$

$$\alpha_{1,1} = \frac{1}{24a} \sum_{i=1}^n (x_{i-1} y_i - x_i y_{i-1}) (2x_{i-1} y_{i-1} + x_{i-1} y_i + x_i y_{i-1} + 2x_i^2 y_i) \quad (11)$$

$$\alpha_{2,0} = \frac{1}{12a} \sum_{i=1}^n (x_{i-1} y_i - x_i y_{i-1}) (y_{i-1}^2 + y_{i-1} y_i + y_i^2) \quad (12)$$

$$\mu_{2,0} = \alpha_{2,0} - \alpha_{1,0}^2 \quad (13)$$

$$\mu_{1,1} = \alpha_{1,1} - \alpha_{1,0} \alpha_{0,1} \quad (14)$$

$$\mu_{0,2} = \alpha_{0,2} - \alpha_{0,1}^2 \quad (15)$$

- Caractéristiques de la bordure : En plus de la longueur de la bordure l , les coordonnées minimales et maximales des pixels de l'objet, x_{min} , x_{max} , y_{min} et y_{max} , ainsi que la distance minimale et maximale du centre de gravité à la frontière, r_{min} et r_{max} , peuvent être calculés à partir de B. Les valeurs minimales et maximales ne sont pas invariantes pour toute transformation. r_{min} et r_{max} sont invariants à la translation et à la rotation, et variant à l'échelle / résolution.
- Normalisation : Les caractéristiques ci-dessus doivent être normalisées pour être utilisées dans une application de la vie réelle à éliminer la translation et la variance d'échelle. Une résolution et un invariant de traduction fonctionnalité x_p peut être calculée comme suit

$$x_p = (x_{max} - x_{cog}) / (x_{cog} - x_{min}) \quad (19)$$

- Dérivés : Dans les entités calculées pour les gestes dynamiques, l'invariance d'un décalage constant peut également être réalisée par dérivation. Calculer la dérivée $f'(t)$ d'une caractéristique $f(t)$ et l'utiliser comme un élément supplémentaire dans le vecteur de caractéristiques pour souligner les changements de $f(t)$ peut être parfois une méthode simple mais efficace pour améliorer les performances de classification.

Les principales caractéristiques géométriques à extraire sont les suivantes :

- Excentricité** : Une mesure possible de l'excentricité e basée sur les moments centraux est donnée.

$$e = (\mu_{2,0} - \mu_0)^2 + 4\mu_{1,1} / a \quad (16)$$

- ii. **Orientation** : L'axe principal de la région est défini comme l'axe du plus petit moment d'inertie. Cette orientation α est donnée par

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right) \quad (17)$$

- iii. **Compacité** : La compacité d'une forme c $[0, 1]$ est définie comme

$$C = \frac{4\pi a}{l^2} \quad (18)$$

Les formes compactes ($c \rightarrow 1$) ont des bordures courtes qui contiennent une grande surface a . Le plus compact la forme est un cercle ($c = 1$), tandis que pour les formes allongées ou effilochées, $c \rightarrow 0$. rotation, translation et invariant d'échelle / résolution.

V. EXPÉRIENCES ET RÉSULTATS

Un aperçu global de notre interface graphique peut être vu dans la figure 3.3

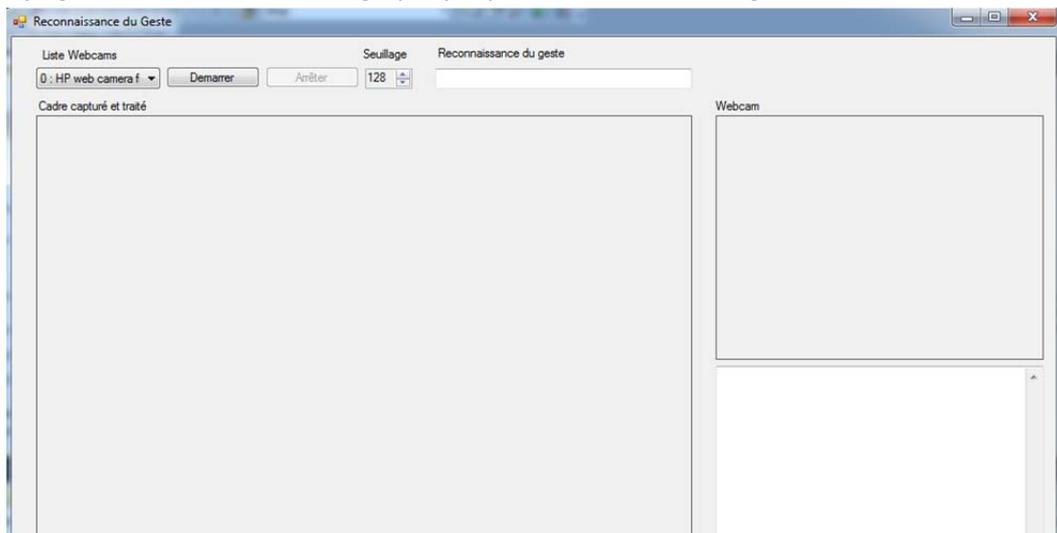


Figure 18 - Interface

- **Liste de webcams**: Affiche les périphériques de capture vidéo détectés sur le système par AForge .Net.

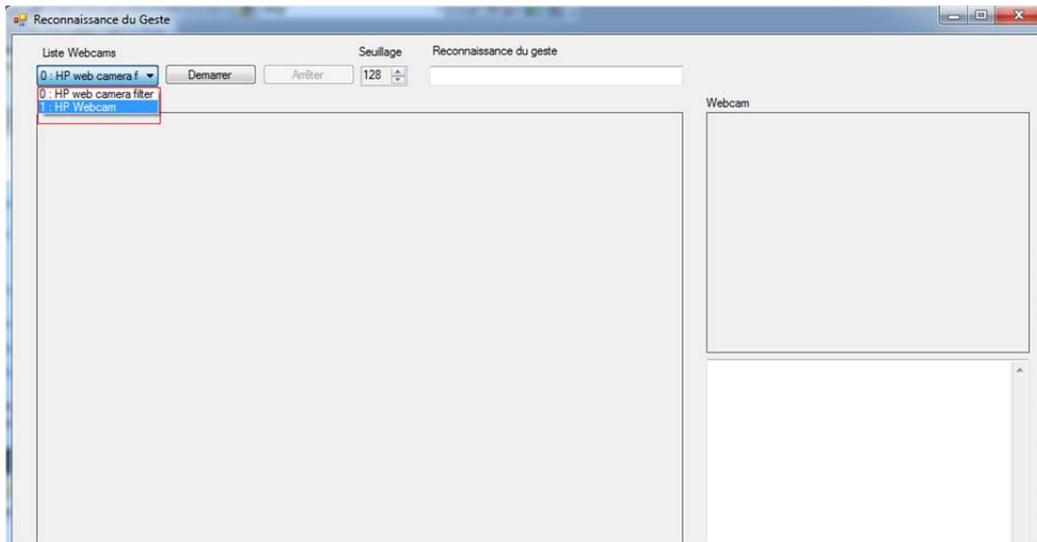


Figure 19 - Choix de la Webcam

- **Boutons de démarrage et d'arrêt:** sont utilisés pour démarrer et arrêter le processus d'acquisition.

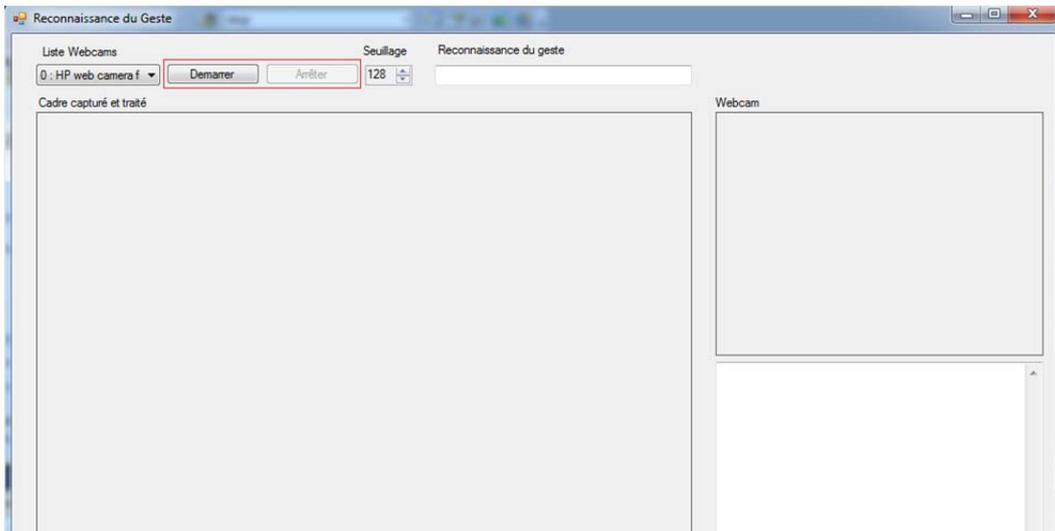


Figure 20 - Bouton « Démarrer & Arrêter »

- **Seuil:** le contrôle est utilisé pour définir le seuil du processus de binarisation permettant à l'utilisateur de s'adapter à différentes conditions.

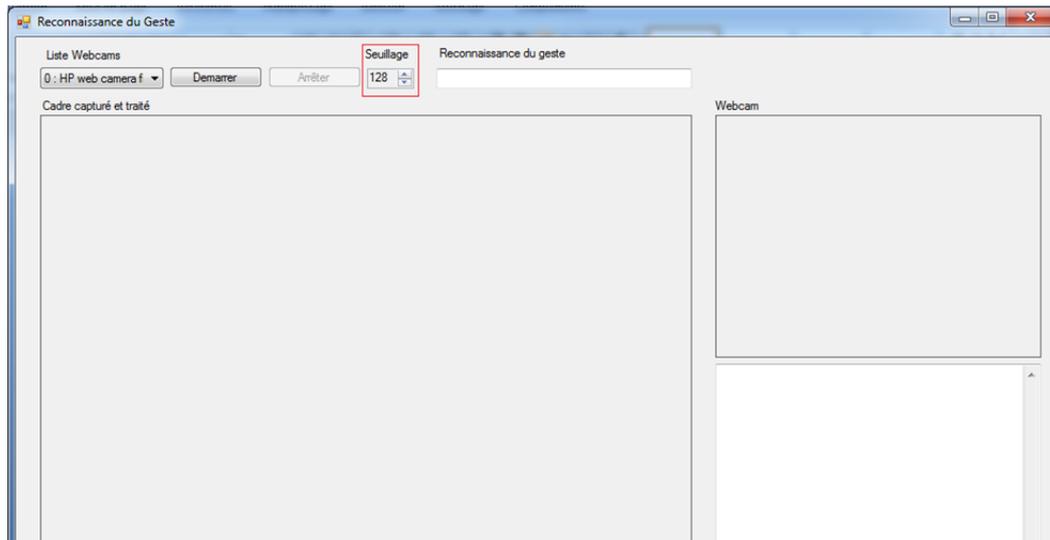


Figure 21 - Choix du Seuil

- **Webcam:** les images capturées à partir de la webcam, binarisées et filtrées sont affichées dans cette zone

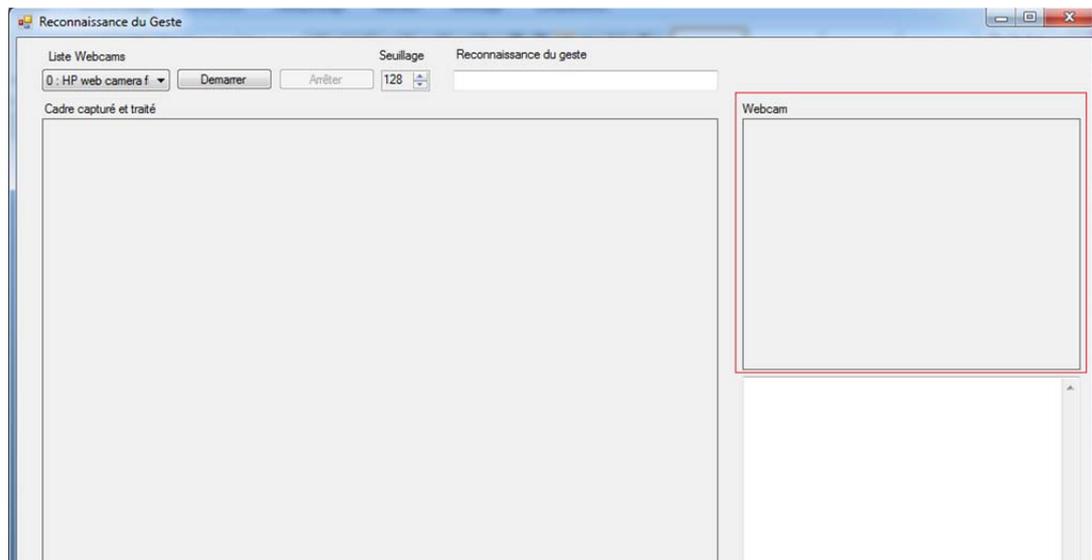


Figure 22 - Images Capturées par la Webcam

- **Cadre capturé et traité :** La zone est utilisée pour afficher les images qui sont capturées à une fréquence d'une image par seconde.

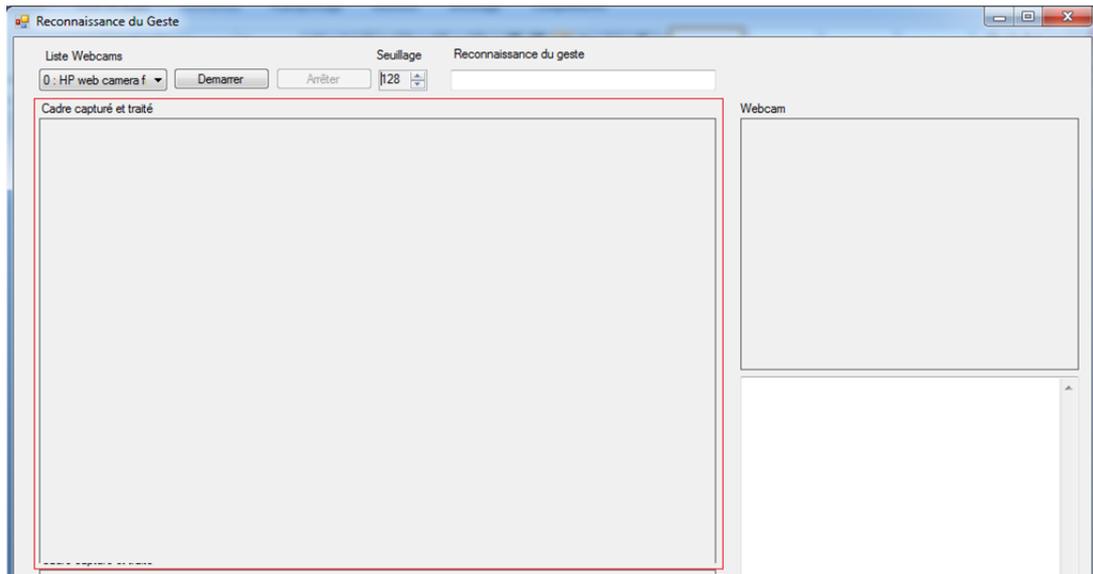


Figure 23 - Cadre capturé & traité

- **Geste reconnu:** la zone de texte permet d'afficher le geste en cours reconnu. Les valeurs sont: Stop, Droite, Gauche, Non reconnu.

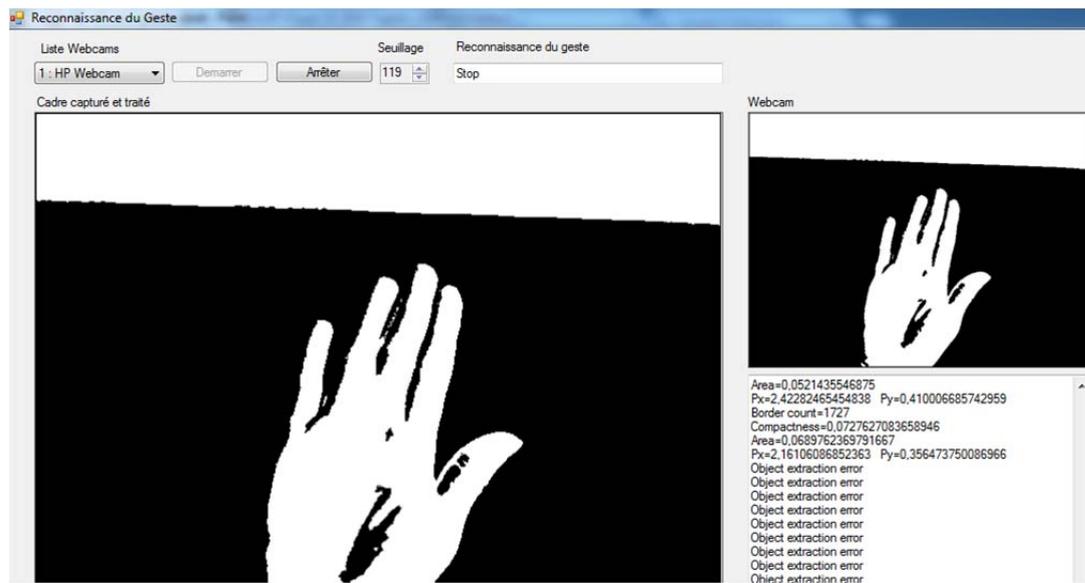


Figure 24 - Geste "STOP"

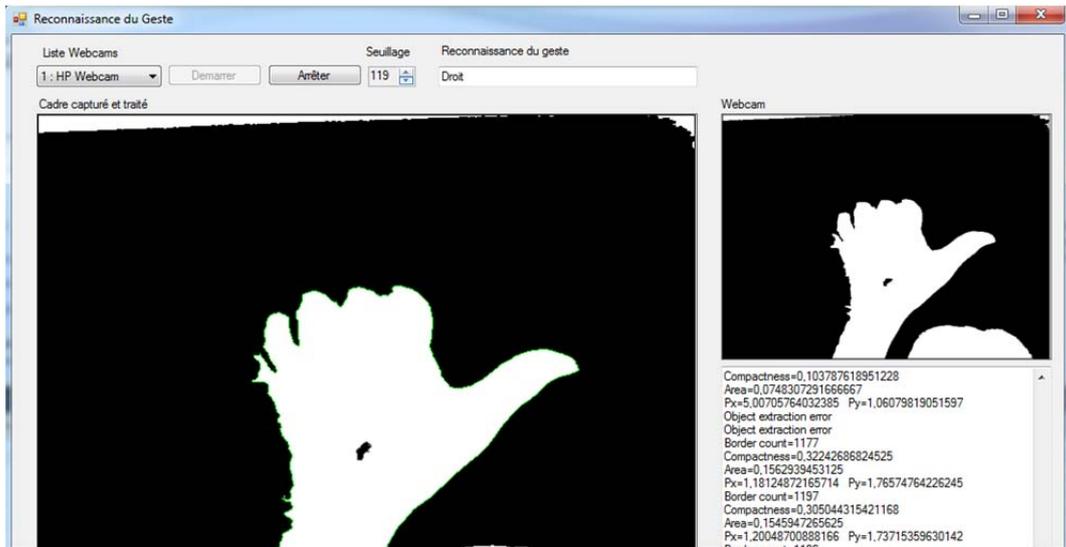


Figure 25 - Geste « Droit »

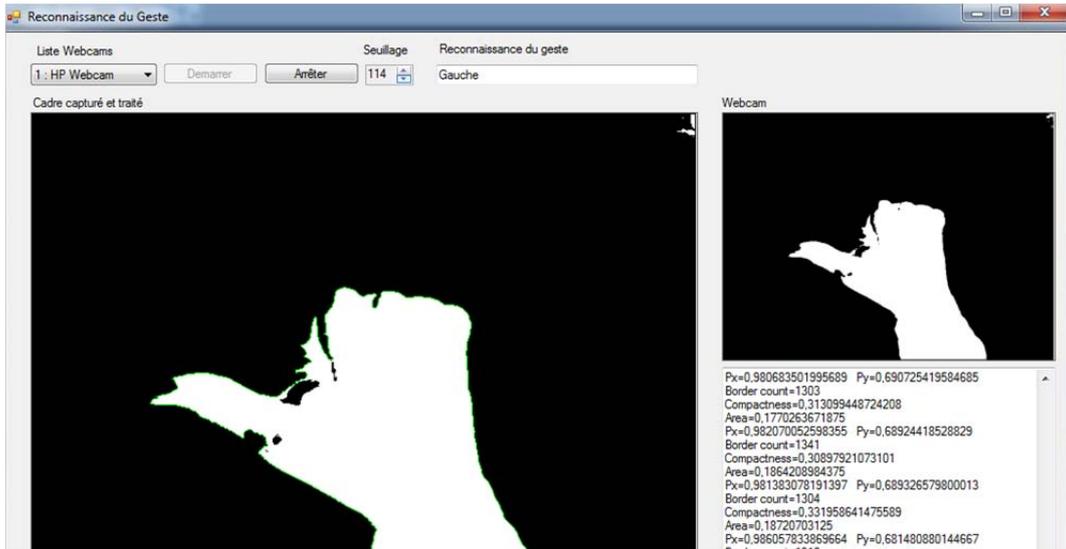


Figure 26 - Geste « Gauche »

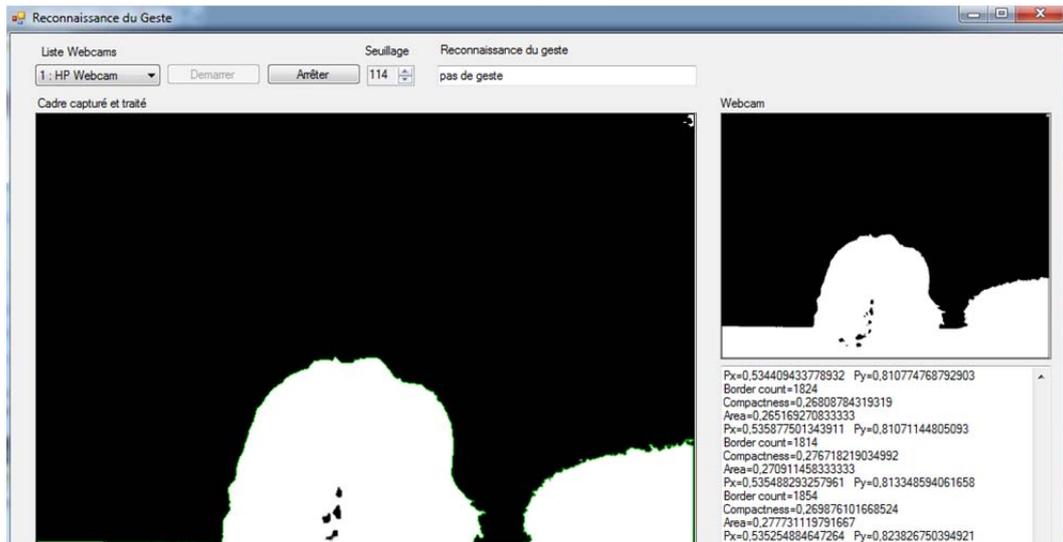


Figure 27 - "Pas de Geste"

- **Zone 1:** la zone de texte permet d'afficher des informations sur le processus en cours et les entités extraites.

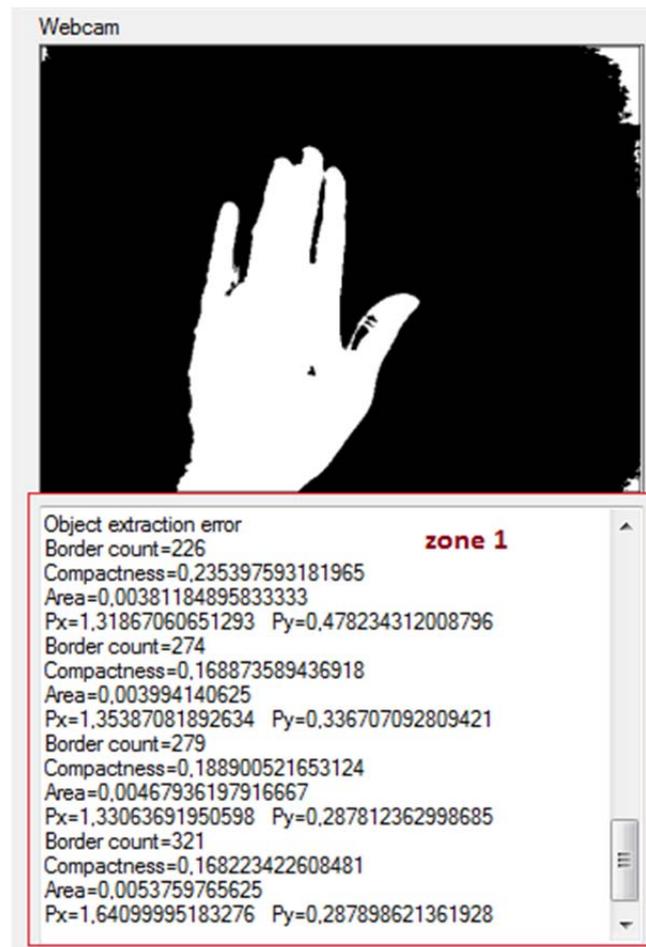


Figure 28 - Zone 1 « Etape Extraction du contour »

VI. **Avantage de l'application**

L'algorithme utilisé fonctionne en temps réel. Ils diffèrent par le contenu du cadre, l'éclairage, les arrière-plans et l'indépendance de l'utilisateur.

1. Neutralité: l'utilisateur parlera sans porter de gants de données, de gants de couleur ou d'autres types de capteurs ou de marqueurs.
2. Auto-adaptation à changer les conditions externes telles que l'éclairage, les arrière-plans. capture de l'installation, contenu de l'image.
3. Caméra: Le matériel et / ou les paramètres de la caméra peuvent changer.
4. Performance comme l'algorithme fonctionne en temps réel pour le monde réel.

VII. **Mode d'utilisation**

L'application a été écrite en C # et nécessite les références AForge: AForge.Controls, AForge.Video et AForge.Video.DirectShow.

Pour exécuter l'algorithme, sélectionnez une webcam dans la liste. Si aucune webcam n'est présente, essayez une autre webcam, réinstallez les pilotes de webcam. Le bouton de démarrage lance le processus d'acquisition, l'image est binarisée, filtrée et affichée dans la zone de la webcam. Une fois par seconde une image est capturée, la région est identifiée, les caractéristiques sont calculées et un geste de la main est identifié, si aucun geste ne peut être trouvé, un message sera affiché. L'arrêt du processus d'acquisition se fait avec le bouton d'arrêt.

Obtenir une bonne capture de la main dépend fortement des conditions d'acquisition. Pour un résultat optimal, la main doit être devant un fond blanc et une illumination suffisante doit être présente pour obtenir un bon contraste de couleurs. Le paramètre de seuil permet de modifier l'étape de binarisation afin que seule l'aiguille soit affichée.

VIII. **Conclusion**

Un système de reconnaissance des gestes de la main, basé sur la vision a été développé. La couleur est utilisée pour le processus de localisation de la main puisque la taille et la forme de la main varient considérablement. Puis un algorithme de description de la région de la main a été utilisé pour trouver les points de bordure de toutes les régions dans une image. Par la suite, un ensemble de caractéristiques géométriques sont ensuite extraites, ces caractéristiques sont l'excentricité, l'orientation, la compacité, la longueur de la bordure, la surface, le centre de gravité (cog), et les moments de second ordre. Enfin, nous avons utilisé le classificateur basé sur des règles pour classer l'extrait caractéristiques au bon signe. Nous pensons avoir atteint notre objectif initial, cependant toute solution peut être améliorée, complétée et optimisée.

Conclusion Générale

Notre projet de thèse porte sur la réalisation d'un système de reconnaissance des gestes de la main, qui regroupe plusieurs étapes par les quelles nous avons essayé d'extraire les meilleures méthodes et effectuer plusieurs testes afin d'avoir un résultat satisfaisant.

Notre système de reconnaissances des gestes de la main basé sur la vision à été implémenté suivant les étapes suivantes

En premier plan nous avons opté pour la webcam pour l'acquisition de la main, nous passons ensuite à la segmentation qui consiste à la localisation de la main par la couleur de peau puisque la taille et la forme de la main varient considérablement. Par la suite le seuillage qui consiste à séparer la main de l'arrière plan et qui se fait manuellement, puis un algorithme de description de la région de la main a été utilisé pour trouver les points de bordure de toutes les régions dans une image. Par la suite, un ensemble de caractéristiques géométriques sont ensuite extraites ces caractéristiques sont l'excentricité, l'orientation, la compacité, la longueur de la bordure, la surface, le centre de gravité (cog). Enfin, nous avons utilisé le classificateur basé sur des règles pour classer le geste selon les trois gestes choisis (stop, droit, gauche). Un taux de reconnaissance d'environ 90% sur les données de test a été atteint

Par ailleurs le critère de la luminosité joue un très grand rôle pour cela nous n'avons pas toujours le résultat souhaité.

Bibliographie

- [1] Muhamme Inayat Ullah Khan, Master Thesis computer Engineering – 2011
- [2] Simon Conseil, thèse de doctorat, Université paul cézanne faculté des sciences et techniques, université paul cézanne aix-marseille iii, Année 2008
- [3] Frédéric Devernay, introduction à la vision par ordinateur
- [4] Berrached Chahrazed, Système de Reconnaissance des gestes de la main, Université Abou Bekr Belkaid, Année 2014
- [5] A. Braffort : Reconnaissance et compréhension de gestes, application à la langue des signes. Thèse de doctorat, Université Paris-XI, 1996.
- [6] C. Cadoz : Le geste canal de communication homme/machine – la communication instrumentale. Techniques et Science Informatiques, 13:31–61, 1994.
- [7] C. Mertz, J. L. Vinot et D. Etienne : Entre manipulation directe et reconnaissance de l'écriture : les gestes écologiques. In Ergonomie et informatique Avancée, Interaction Homme-Machine Ergo-IHM, p. 145–152, 2000.
- [8] P. A. Harling et A. D. N. Edwards : Hand tension as a gesture segmentation cue. In Proc. of Gesture Workshop, 1996.
- [9] 9789814585682-c1 with gloves.pdf « Historical Development of Hand Gesture Recognition »
- [10] S. C. W. Ong et S. Ranganath : Automatic sign language analysis : A survey and the future beyond lexical meaning. IEEE trans. on Pattern Analysis and Machine Intelligence, 27(6):873–891, juin 2005.
- [11] T. Starner et A. Pentland : Visual recognition of American sign language using hidden Markov models. In IEEE Int. Work. on Automatic Face and Gesture Recognition, p. 189–194, 1995.
- [12] T. Starner, J. Weaver et A. Pentland : Real-time american sign language recognition using desk and wearable computer based video. IEEE trans. on Pattern Analysis and Machine Intelligence, 20(12):1371–1375, 1998.
- [13] J. Segen et S. Kumar : Fast and accurate 3D gesture recognition interface. In Proc. of the IEEE Int. Conf. on Pattern Recognition, 1998.
- [14] N. Otsu : A threshold selection method from gray level histograms. IEEE trans. on Systems, Man, and Cybernetics, 9:62–66, mars 1979.

[15] F. K. H. Quek : Toward a vision-based hand gesture interface. In Proc. Of the Virtual reality software and technology conf., p. 17–31, 1994.

[16] Sonka M, Hlavac V, Boyle R. Image Processing, Analysis and Machine Vision. Brooks Cole; 1998.

[17] Kraiss K. Advanced Man-Machine Interaction Fundamentals and Implementation. Springer; Berlin; 2006.

[18] Alaa el-din mohamed Riad & Samaa Shohieb, 1Faculty of Dean, Faculty of Computers and Information Sciences, Faculty, Mansoura University, Egypt, Published 3rd February 2014

[19] Zieren J, Kraiss KF. Robust Person-Independent Visual Sign Language Recognition. In Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis IbPRIA 2005, volume Lecture Notes in Computer Science; 2005.

[20] Aforrge. Net; he open source fra Available: <http://www.aforgenet.com>

- **Résumé**

Ce projet repose sur la reconnaissance des gestes de la main et s'inspirent des domaines de l'intelligence artificielle qui est un domaine important de la recherche. De tels systèmes sont considérés comme des outils utiles pour aider les sourds d'une part et également utilisés pour contrôler des ordinateurs, des écrans, des pointeurs ou des dispositifs de sélection à base de caméra, comme ceux utilisés sur les «Smart TV» modernes ou jeux de console qui utilisent la caméra Microsoft Xbox Kinect.

Ce travail est réalisé par C # et utilise le framework AForge.Net pour capturer des images à partir d'une webcam en temps réel et ceci se fait en quatre grandes étapes la segmentation en utilisant la méthode RGB, le seuillage avec des valeurs inférieur et supérieur, l'extraction des caractéristiques géométriques et enfin le classificateur pour déterminer le geste.

Mots clés : reconnaissance des gestes, interaction homme machine, vision par ordinateur, segmentation, seuillage, contour, codage, classification.

- **Abstract**

This project is based on the recognition of hand gestures and is inspired by the fields of artificial intelligence, which is an important area of research. Such systems are considered useful tools to help the deaf on the one hand and also used to control computers, screens, pointers or camera-based selection devices, such as those used on modern smart TVs. or console games that use the Microsoft Xbox Kinect camera.

This work is done by C # and uses the AForge.Net framework to capture images from a webcam in real time and this is done in four main steps segmentation using the RGB method, the thresholding with lower values and superior, extraction of geometric features and finally the classifier to determine the gesture.

keywords

gesture recognition, human machine interaction, computer vision, segmentation, thresholding, contour, coding, classification.

ملخص

يستند هذا المشروع على الاعتراف بحركات اليد ، وهو مستوحى من حقول الذكاء الاصطناعي ، وهو مجال هام من مجالات البحث. وتعتبر هذه الأنظمة أدوات مفيدة لمساعدة الصم من ناحية، كما تستخدم أيضاً في التحكم في أجهزة الكمبيوتر، الشاشات، المؤشرات أو أجهزة التحديد القائمة على الكاميرا، مثل تلك المستخدمة على أجهزة التلفزيون الذكية الحديثة. أو ألعاب وحدة التحكم Microsoft Xbox Kinect التي تستخدم كاميرا

لالتقاط الصور من كاميرا الويب في الوقت الحقيقي ويتم ذلك في تجزئة AForge.Net ويستخدم إطار C # يتم هذا العمل بواسطة ، العتبة ذات القيم الأقل و متفوقة ، واستخراج الميزات الهندسية وأخيراً المصنف RGB أربع خطوات رئيسية باستخدام طريقة لتحديد الإيماءة

الكلمات الرئيسية

التعرف على الإيماءات ، تفاعل الآلة البشرية ، رؤية الكمبيوتر ، التجزئة ، العتبة ، الكفاف ، الترميز ، التصنيف