

MA-004-149-1

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des Sciences

Département d'Informatique



MEMOIRE DE MASTER

Option : Génie en Système Informatique

CONCEPTION ET DEVELOPPEMENT D'UNE
INTERFACE UTILISATEUR A BASE DE
RECONNAISSANCE GESTUELLE POUR MOBILES

Par

Ishak AMRI

Mohamed SNOUSSI

Promoteur : M. Mohamed Redha SIDEMOU

Encadreur : M. Djamel ZAHAL

2012 / 2013

MA-004-149-1

RESUME

Les modes d'interactions homme/machine ne cessent de changer grâce à la technologie qui est en constante évolution et l'intelligence artificielle joue un grand rôle dans ces nouvelles méthodes d'interactions entre l'homme et la machine.

Les smartphones ou téléphones intelligents sont aujourd'hui omniprésents dans notre quotidien. Plus son utilisation est facilitée et mieux le smartphone s'insère dans notre vie et son utilisation devient plus efficace.

Les interactions à base de « touch » ont révolutionné les usages et notre sujet porte sur une autre possibilité d'interaction basée sur la reconnaissance de geste.

La reconnaissance de geste est la capacité pour un appareil mobile de reconnaître le geste capturé avec la caméra, de l'interpréter afin de réaliser l'action adéquate.

Mots-clés : Reconnaissance gestuelle, Intelligence artificielle, Détection, Suivi à la trace, Contexte mobile.

ABSTRACT

The interaction between human and machines is expanded with the evolution of technology, so is the artificial intelligence of those machines which allows a higher level of comprehension between the two of them.

The mobile has become a smart device, and it is an important part of many people's life. The easier its manipulation is made, the better it fits into our lives and its use becomes more efficient.

Many works were achieved from user interfaces to touch screens. In our work we explore yet another possibility, that is camera-based gesture recognition, and we will try to implement it into the mobile world.

Gesture recognition is the ability for a mobile device to recognize the gesture captured with its frontal camera to interpret it in order to realize the suitable action.

Keywords: Gesture recognition, Artificial intelligence, Detection, Tracking, Mobile context.

ملخص

وسائط التفاعل بين الإنسان والآلة تتغير باستمرار مع التطور التكنولوجي المشهود حيث أن للذكاء الاصطناعي الدور الحاسم في ازدهار هذه الطرق الجديدة للتواصل.

الهواتف الذكية تعتبر جزءا من حياتنا اليومية. فكلما سهل استخدامها، زاد انتشارها وزاد اقبال الناس عليها.

فالهواتف ذات شاشات اللمس قد أحدثت ثورة في عالم الجوال، وهذا ما دفعنا لتناول هذا المشروع الذي يهدف الى دراسة كيفية جديدة للتعامل، ألا وهي خاصية التعرف على الإيماءات. هذه القدرة تأهل الجهاز المحمول على فهم الإيماءات التي يحدثها المستخدم امام الكاميرا، ليقوم بعد ذلك بتنفيذ الإجراءات المناسبة.

الكلمات الرئيسية : التعرف على الإيماءات، التعرف، التعقب، سياق المحمول.

REMERCIEMENTS

On tient tout d'abord à remercier vivement notre promoteur M. Mohamed Redha SIDEMOU, d'avoir accepté de nous encadrer, ainsi que pour ses remarques constructives et ses recommandations.

On remercie tout particulièrement notre encadreur M. Djamel ZAHAL, Fondateur de DzMob pour le temps qu'il nous a consacré et pour ses conseils lumineux qui ont été d'une grande utilité.

On adresse notre grande reconnaissance à tous les enseignants qui ont donné de leurs mieux pour notre formation durant ces cinq ans d'études et qui ont su nous faire apprécier l'informatique.

Enfin, on remercie de tout notre cœur M. Zakaria BELHADJ pour son soutien ainsi que son présence inestimable à nos côtés durant toute la durée de ce travail.

TABLE DES MATIERES

INTRODUCTION GENERALE	9
CHAPITRE 1 : ETAT DE L'ART	12
1.1. Introduction	12
1.2. Application de la gestuelle aujourd'hui	13
3.1.1. Kinect.....	13
3.1.2. Samsung Galaxy S4	15
1.3. Techniques de vision pour la reconnaissance gestuelle :.....	16
1.3.1. La Détection.....	16
1.3.2. Le suivi à la trace	21
1.3.3. La reconnaissance	26
1.3.4. Conclusion	28
CHAPITRE 2 : APPRENTISSAGE AUTOMATIQUE : Apprentissage à partir d'exemple .	30
2.1. Introduction	30
2.2. Apprentissage à partir d'exemples : l'algorithme « \$1 Recognizer »	30
2.2.1. Définition	31
2.2.2. Caractéristiques.....	31
2.2.3. Algorithme	31
2.2.4. Limitation de l'algorithme \$1 :.....	34
2.2.5. Améliorations adaptés.....	34
2.3. Conclusion.....	35
CHAPITRE 3 : CONCEPTION	36
3.2. Introduction	36
3.3. Détection	38
3.3.1. Exemple	45
3.4. Suivi à la trace (ou tracking)	45

3.4.1. La méthode « Second chance ».....	48
3.5. Reconnaissance	49
3.6. Exemple complet.....	51
3.7. Conclusion.....	53
Chapitre 4 : Implémentation	54
4.1. Introduction	54
4.2. Application de test pour le débogage	54
4.3. Application concrète	55
4.4. Les outils de travail	58
4.4.1. Flex	58
4.4.2. Actionscript.....	58
4.4.3. SVN	58
4.4.4. Trello.....	59
4.5. Conclusion.....	59
CONCLUSION GENERALE.....	60
Bibliographie	62
Glossaire	70

LISTE DES FIGURES

Figure 1.1 : Capture d'un corps humain par la Kinect.	14
Figure 1.2 : L'interface gestuelle du Galaxy S4.	15
Figure 1.3 : Exemple d'application du filtre particulaire dans la détection d'une main.	24
Figure 2.1 : Différences de temps considérables et des chiffres résultant de points.	32
Figure 2.2 : Un geste « étoile » échantillonné à $N = 32, 64$ et 128 points.	33
Figure 2.3 : Rotation d'un triangle de sorte que son « angle indicative » est à 0°	33
Figure 3.1 : Architecture du système.	36
Figure 3.2 : Flux d'image acquis à partir de la caméra.	39
Figure 3.3 : Couplage d'images.	39
Figure 3.4 : Application d'une différence entre deux images.	40
Figure 3.5 : Augmentation de contraste d'une image.	41
Figure 3.6 : Filtre flou appliqué sur une image.	41
Figure 3.7 : Zone de délimitation encadrée.	44
Figure 3.8 : Détection de la main dans un geste complet.	45
Figure 3.9 : Suivi à la trace d'un geste complet.	46
Figure 3.10 : Trajectoire parcourue par une main.	46
Figure 3.11 : Un exemple qui schématise un suivi dans un cas normal.	47
Figure 3.12 : Un exemple qui schématise un suivi incomplet.	47
Figure 3.13 : Un exemple qui schématise la méthode « second chance ».	48
Figure 3.14 : Reconnaissance d'une trajectoire de mouvement.	50
Figure 3.15 : Un exemple d'une reconnaissance d'un geste complet. (Partie 1).....	51
Figure 3.16 : Un exemple d'une reconnaissance d'un geste complet. (Partie 2).	52
Figure 4.1 : Capture d'écran de notre application sur tablette mobile pour la reconnaissance gestuelle à base de caméra.	54
Figure 4.2 : Différents gestes pour naviguer dans l'application.	56
Figure 4.3 : Un geste de gauche à droite pour aller à l'album suivant	56
Figure 4.4 : Un geste de bas au haut pour accéder à l'album.	57
Figure 4.5 : Un geste de gauche à droite pour aller à l'image suivante de l'album	57
Figure 4.6 : Un geste en agitant la main pour sortir de l'album	57

INTRODUCTION GENERALE

Aujourd'hui, les téléphones mobiles également appelé téléphones portables ou téléphones cellulaires, sont devenus un élément essentiel de notre vie quotidienne. Tout le monde a un téléphone cellulaire personnel. L'amélioration de la technologie a fait un grand changement dans l'histoire de la téléphonie mobile, transformant les énormes briques comme les téléphones mobiles de 1995 à smartphones épuré et élégant que nous portons avec nous maintenant. Les analystes prévoient qu'en 2020, plus de 10 milliards d'appareils mobiles seront connectés à internet, contre 2 milliards aujourd'hui [1]. C'est au travers de cette tendance que nous assistons à une fragmentation de ce qu'une première révolution avait popularisé dans les entreprises à la fin des années 70' : l'ordinateur personnel. En 2007, Apple a dévoilé l'iPhone d'Apple, qui a été le premier smartphone à écran tactile avancé au monde. Qui comporte son système d'exploitation « iOS », et en permettant aux applications de fonctionner sur le téléphone, il avait permis à des téléphones cellulaires à devenir l'appareil mobile primaire d'utilisation [2]. Plus l'utilisation du mobile est facilitée et mieux il s'insère dans notre vie et son utilisation devient plus efficace.

Les interactions homme/machine ne cessent de changer grâce à la technologie qui est en constante évolution et l'intelligence artificielle joue un grand rôle dans ces nouvelles méthodes d'interactions entre l'homme et la machine. Dans cette optique, la recherche en reconnaissance de gestes s'est attelée peu à peu au développement de systèmes capables de reconnaître les gestes de l'homme et de les interpréter afin d'enrichir l'expérience utilisateur. La reconnaissance de gestes est une tâche complexe impliquant divers aspects dont la modélisation et l'analyse de mouvement [3].

Le nombre croissant de dispositifs mobiles les place au sommet de l'interaction homme-machine, ils ne sont pas conçus pour les mêmes usages qu'un ordinateur classique, ils sont en effet utilisés la plupart du temps dans des conditions de mobilité où l'utilisateur ne dispose pas du même temps de réalisation de ses tâches.

De plus, les facteurs de forme et les modes d'entrée des dispositifs mobiles sont différents (taille de l'écran, clavier restreint, stylet, ...), ce qui rend l'interaction plus complexe.

L'intégration de nouveaux capteurs (accéléromètre, gyroscope, caméra, boussole, ...), permettant de détecter les mouvements et orientations du mobile ou de l'utilisateur qui les manipule, apporte de nouvelles possibilités d'interaction fondées sur les gestes que l'utilisateur réalise avec son dispositif.

Les gestes sont très présents dans nos vies, d'après Axtell « *Sans gestes, notre monde serait statique, incolore...* » [4]. Mario Pei, un expert de communication, explique que les gens peuvent produire jusqu'à 700,000 signes physiques différents, le visage seul est capable de produire 250,000 expressions [5].

Ces découvertes numériques indiquent que les gestes peuvent jouer un rôle significatif dans l'Interaction Homme Machine (IHM), Ils expriment aussi la difficulté énorme pour des chercheurs IHM pour concevoir et développer les techniques d'interaction à base de geste qui sont efficaces.

Vu que les interactions à base de "touch" ont révolutionné les usages du téléphone mobile, notre sujet porte sur une autre possibilité d'interaction basée sur la reconnaissance de geste en utilisant le dispositif « caméra ». La reconnaissance de geste est la capacité pour un dispositif mobile de reconnaître le geste capturé avec la caméra, de l'interpréter afin de réaliser l'action adéquate.

Après plusieurs recherches sur le domaine de la reconnaissance du geste à base de la caméra, nous avons pu conclure et décomposer notre projet sur 3 phases, qui seront les piliers sur lesquels se tient notre système :

- **Détection** dans laquelle on détecte la main de l'utilisateur filmée par la caméra.
- **Tracking (ou le suivi à la trace)** pour suivre le mouvement de la main.
- **Reconnaissance**, l'étape qui permet d'interpréter le mouvement, de lui donner un sens afin de le faire correspondre à une action particulière sur le dispositif mobile.

Pour chacune de ses étapes, nous avons étudié l'état de l'art de nombreux chercheurs ainsi que les techniques qu'ils ont implémenté dans leurs travaux. Nous avons ensuite adapté ces méthodes au contexte mobile tout en améliorant et ajoutant des méthodes que nous avons créées après nos différentes recherches.

Le premier problème confronté est la complexité des algorithmes, étant le mobile peu puissant par rapport aux machines classiques. La plus part des algorithmes nécessitent beaucoup de traitement et le mobile n'est pas adapté. Nous allons veiller à utiliser la bonne approche pour ne pas pénaliser les performances sur un téléphone et rendre l'interprétation des gestes aussi fluide que possible. Pour cela nous utilisons et développons des méthodes optimisées côté mobiles en prenant en considération les ressources matérielles limitées existantes.

Autre contrainte, le mouvement perpétuel d'un mobile dans la main de son utilisateur peut accroître la difficulté de reconnaître une main dans un environnement en mouvement. Nous avons essayé d'éliminer les mouvements indésirables et insignifiants autant que possible ce qui sera le cœur de notre travail.

Étant donné que notre étude est basée sur la reconnaissance des gestes sur un dispositif mobile, il est très important de définir comment l'appareil détecte la main, comment faire suivre sa trajectoire, et comment l'interpréter afin de déduire le geste. C'est ce qu'on va tenter d'y apporter des réponses à travers notre étude organisés en cinq chapitres comme suit :

Chapitre 1 : dans ce chapitre, nous allons explorer le monde de la reconnaissance gestuelle et nous présentons les travaux existants sur ce thème, qui sera sous forme d'un état de l'art, et mettra en évidence nos avis en prenant en considération le contexte mobile.

Chapitre 2 : dans ce chapitre, nous présentons le principe, les concepts fondamentaux et les améliorations adaptées sur l'algorithme \$1, utilisé dans le système d'apprentissage automatique de notre projet.

Chapitre 3 : il est consacré à la présentation des différents concepts et techniques développés pour notre approche afin d'intégrer la reconnaissance gestuelles à base de la caméra frontale dans les appareils mobiles. Ainsi des explications approfondies des 3 principales étapes du processus.

Chapitre 4 : il est dédié à la présentation de l'environnement de développement que nous avons utilisé pour valider notre travail.

CHAPITRE 1 : ETAT DE L'ART

1.1. Introduction

En quelques années, la multiplication et la diversité des dispositifs mobiles dans nos vies quotidiennes ont entraîné de nouveaux défis pour les concepteurs, De nombreuses techniques d'interaction ont été proposées dont celle des dispositifs mobiles qui en est un enjeu majeur de l'IHM.

Cette section présente un état de l'art, on aborde la question de l'interaction gestuelle sur dispositif mobile. Ces dispositifs, à présent communs, se distinguent des ordinateurs conventionnels principalement par leurs périphériques d'interaction avec l'utilisateur (écrans de taille restreinte mais tactiles, capteurs divers tels que les accéléromètres, ...) ainsi que par le contexte dans lequel ils sont utilisés.

Dans ce chapitre, notre travail sera une exploration du vaste domaine des techniques d'interaction homme machine en général et plus précisément l'interaction sur des dispositifs mobiles.

C'est pourquoi ce chapitre se compose de deux parties, la première sera une vue globale sur l'application de la gestuelle aujourd'hui.

La deuxième partie de ce chapitre est un état de l'art sur les différentes techniques qui nous intéressent sur le domaine de la reconnaissance gestuelle basée sur la main à partir d'une caméra, les algorithmes utilisés, et les technologies existantes, qui vont nous aider à implémenter ces techniques dans des dispositifs mobiles.

Le travail de Myron Krueger [6] sur la réalité artificielle au début des années 1980 peut être considéré comme la première introduction à l'interaction gestuelle dans la réalité artificielle avec des grandes images projetées en utilisant une caméra vidéo pour le dépistage de tout le corps.

1.2. Application de la gestuelle aujourd'hui

Deux des adeptes du concept de la reconnaissance gestuelle à base de camera sont Microsoft avec la Kinect et Samsung avec son Galaxy S4.

Le premier est basé sur la manipulation directe d'objets représentés en trois dimensions. L'utilisateur est plongé dans un monde virtuel, fonctionnant comme le monde réel, mais dans lequel l'ordinateur interprète toutes ses actions comme des commandes. Tous cela en gardant le dispositif fixe. Tandis que le Samsung Galaxy S4 est un appareil mobile qui interprète uniquement les gestes de la main.

3.1.1. Kinect

1.2.1.1 Définition :

Kinect est un périphérique de détection de mouvement développée par Microsoft pour la console de jeu vidéo Xbox 360 et PC Windows. C'est un module complémentaire périphérique pour la console Xbox 360, il permet aux utilisateurs de contrôler et d'interagir avec la Xbox 360 sans avoir besoin de toucher un contrôleur de jeu (manette), grâce à une interface utilisateur naturelle en utilisant des gestes et des commandes vocales. [7]

- **Capteur de mouvement :**

Kinect utilise un capteur de mouvement qui capte notre corps tout entier. C'est pour cela que lorsqu'on joue, il n'y pas qu'une détection des mains et des pouces.

- **Dépistage Squelettique :**

Lorsque l'on joue, Kinect crée un squelette numérique de notre corps basé sur des données de profondeur, c'est à dire en trois dimensions. Ainsi, lorsque nous nous déplaçons à gauche ou à droite ou à sauter partout, notre capteur traite les mouvements et les traduit dans le jeu. [8]

1.2.1.2 Système Kinect :

La Kinect utilise la lumière structurée et l'apprentissage automatique, afin d'Inférer la position du corps. Le processus se fait en deux étapes, d'abord on calcule la

profondeur (à l'aide de la lumière structurée), puis on conclut la position du corps (en utilisant l'apprentissage machine).

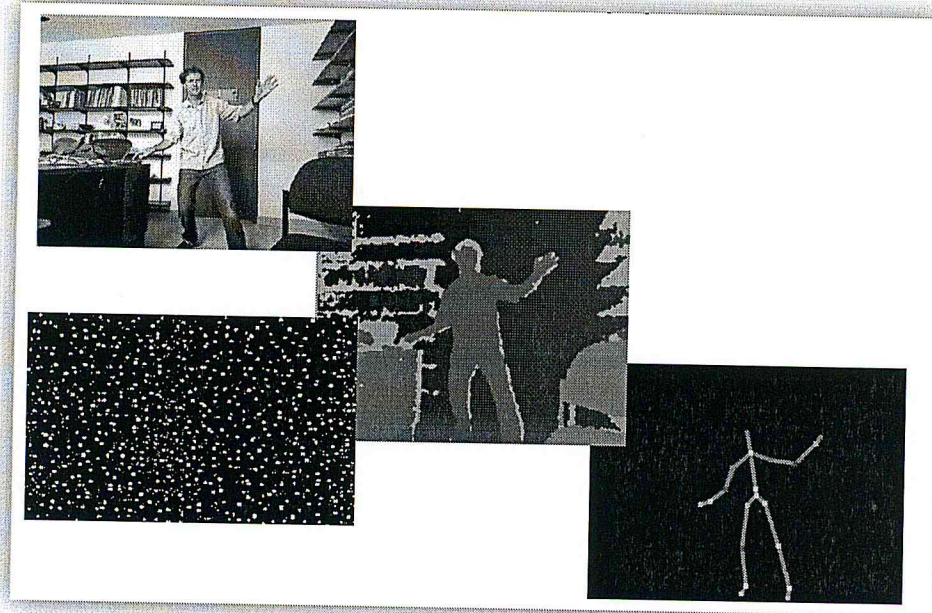


Figure 1.1 : Capture d'un corps humain par la Kinect.

Une fois que l'appareil dispose de données à traiter, le processeur les utilise pour le faire fonctionner, Il distingue les visages, et d'autres aspects de l'environnement de la zone, ainsi il reconnaît les gens et se souvient quand ils reculent dans la pièce et filtre le bruit dans la chambre. Le système fait tout cela en continu à 30 images par seconde. [9]

1.2.1.3 Contrainte :

Il y a beaucoup de contraintes environnementales que la Kinect impose. A l'intérieur de sa documentation distribuée, la réduction de la lumière solaire ambiante dans la pièce est un point souligné.

La deuxième contrainte plus importante est la taille d'espace de jeu (la pièce), nous avons besoin d'une grande pièce pour utiliser la Kinect. Microsoft note que nous devrions mettre le capteur entre deux (0.6 mètre) à six pieds (1.8 mètre) de la terre et centré avec notre télévision. [10]

3.1.2. Samsung Galaxy S4

La fameuse entreprise Samsung a adopté récemment le système de reconnaissance gestuelle à base de camera dans son nouvel smartphone le Galaxy S4, l'utilisateur peut interagir avec (contrôler) son appareil mobile par des gestes basiques avec ses main devant la caméra frontale. Sous le nom de « *air gesture* », les gestes sont démontrés dans la figure 1.2 ci-dessous :

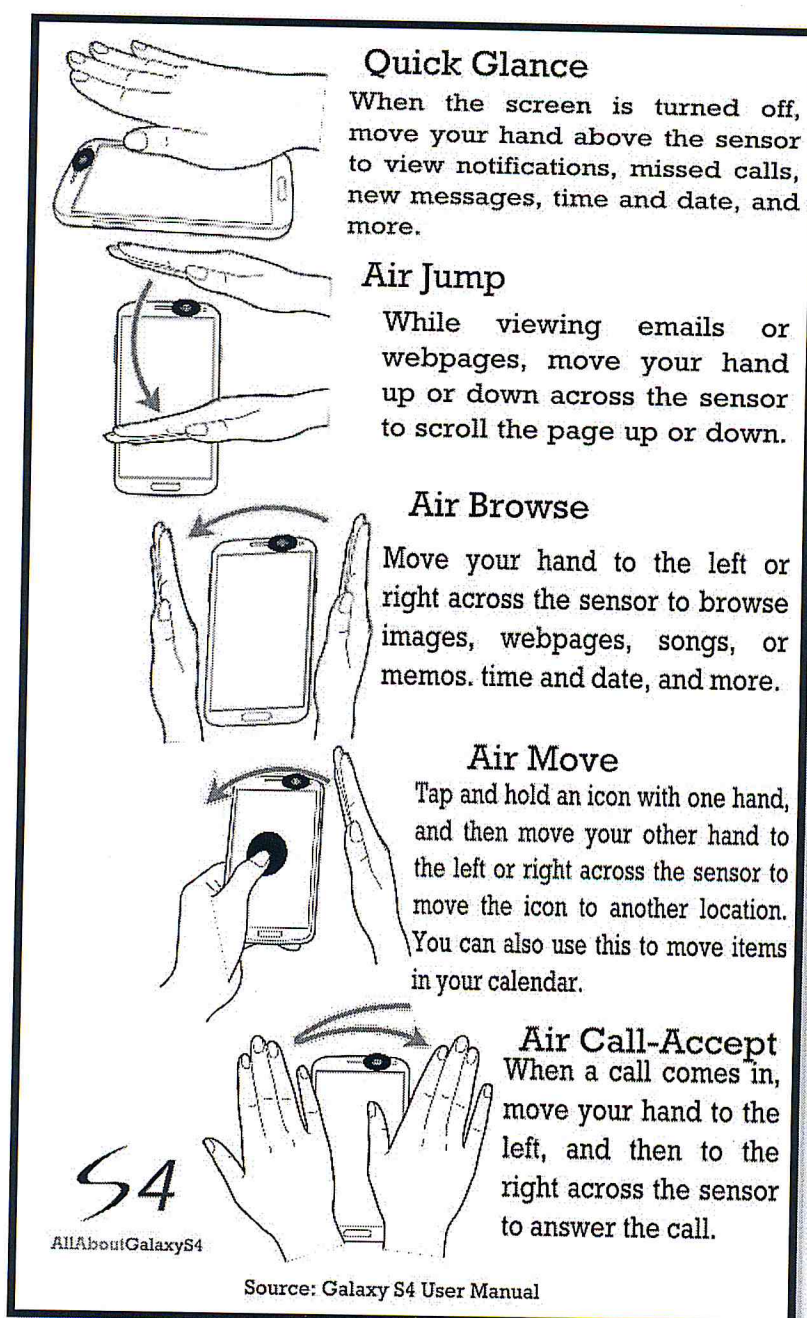


Figure 1.2 : L'interface gestuelle du Galaxy S4. [84]

1.3. Techniques de vision pour la reconnaissance gestuelle :

La plupart des systèmes interactifs à base de la main peut être constitué de trois couches: la détection, le suivi (*tracking*) et la reconnaissance. La couche de détection est chargée de définir et extraire des caractéristiques visuelles qui peuvent être attribuées à la présence de mains dans le champ de vision de la caméra. La couche de suivi est chargée de l'exécution temporelle des données d'association entre les images successives, de sorte que, à chaque moment dans le temps, le système puisse être conscient de « Où est qui ? ». Enfin, la couche de reconnaissance est chargée de regrouper les données spatio-temporelles extraites dans les couches précédentes et l'affectation des groupes qui en résultent avec les étiquettes associées à des catégories particulières de gestes.

1.3.1. La Détection

La première étape dans les systèmes de reconnaissance de gestes est la détection de la main qui sera ensuite utilisée dans le suivi à la trace et enfin l'étape de reconnaissance.

Un grand nombre de méthodes ont été proposées dans la littérature qui utilise plusieurs types de caractéristiques visuelles et, dans de nombreux cas, leur combinaison. Ces caractéristiques sont : la couleur de peau, la forme, le mouvement.

1.3.1.1. Couleur de peau

La segmentation basée sur la couleur de la peau a été utilisée dans plusieurs approches pour la détection des mains. Une décision importante doit être prise afin de fournir le modèle de couleur de la peau, c'est à dire le choix de l'espace couleur à utiliser. Plusieurs espaces colorimétriques ont été proposées, y compris RGB, RGB normalisé, HSV, YCrCb, YUV, etc. Les espaces colorimétriques qui séparent efficacement la chromaticité des composantes de la luminance de la couleur sont généralement jugés préférables.

Quelques méthodes comme dans [11] et [12], existent pour augmenter l'invariance contre la variabilité d'illumination, Ceux qui fonctionnent dans le HSV [13], YCrCb [14], ou YUV [15] chromatiques, afin de rapprocher *la chromaticité* de la peau plutôt que sa valeur de couleur apparente. Ces méthodes éliminent généralement la composante de luminance, pour

éliminer l'effet d'ombre et les changements d'éclairage. Le vecteur de couleur 2D restant est presque constant pour les régions de la peau et un histogramme 2D des pixels d'une région contenant la peau présente un pic intense à la couleur de la peau. Les régions où cette probabilité est supérieure à un seuil sont détectées.

Plusieurs méthodes [13], [16] et [15] utilisent les distributions de couleur pré-calculées extraites de l'analyse statistique de grands ensembles de données. Par exemple, dans [16], un modèle statistique de la couleur de peau a été obtenu à partir de l'analyse de milliers de photos sur le Web. En revanche, des méthodes telles que celles décrites dans [17] sont de construire un modèle de couleur basée sur des échantillons prélevés de couleur de peau lors de l'initialisation du système.

La couleur perçue de la peau humaine varie considérablement selon les genres humains ou même entre individus du même genre. Une variabilité supplémentaire peut être introduite en raison de l'évolution des conditions d'éclairage et / ou des caractéristiques de caméra. Par conséquent, les approches fondées sur la couleur pour la détection des mains ont besoin d'employer des moyens pour compenser cette variabilité. Cependant, plusieurs méthodes sont proposées mais restent sensibles aux changements rapides, ou des conditions d'éclairage mixtes.

En général, la segmentation couleur peut être confondue par des objets de fond qui ont une distribution de couleur semblable à la peau humaine. Une façon de régler ce problème est basée sur la soustraction du fond [18]. Toutefois, la soustraction de fond est généralement basée sur l'hypothèse que le système de caméra ne se déplace pas par rapport à un arrière-plan statique.

La couleur de peau n'est qu'un des nombreux indices qui seront utilisés pour la détection de la main. Par exemple, dans les cas où les visages apparaissent également dans le champ de la caméra, un traitement supplémentaire est nécessaire pour distinguer les mains du visage [19] [20]. Ainsi, la couleur de peau a été utilisée en combinaison avec d'autres indices pour obtenir de meilleures performances.

1.3.1.2. Forme

La forme caractéristique des mains a été utilisée pour les détecter sur les images de plusieurs façons. Beaucoup d'informations peuvent être obtenues seulement par l'extraction des contours des objets dans l'image. S'il est correctement détecté, le contour qui représente la forme de la main ne dépend pas directement de la couleur de peau et l'éclairage.

Dans les systèmes de tirage d'2D/3D [6] [21], la main de l'utilisateur est directement extraite en tant que contour en supposant un fond uniforme et en effectuant une détection de bord en temps réel à cette image. Des exemples de l'utilisation de contours comme caractéristiques se retrouvent à la fois dans le modèle [22] et les techniques basées sur l'apparence [18]. Dans [23] les doigts et le bras sont sélectionnés par le regroupement des séries de bords parallèles.

Des descripteurs topologiques locaux ont été utilisés pour faire correspondre un modèle avec les bords de l'image. Dans [24], le descripteur de contexte de forme est proposé, ce qui caractérise un emplacement de point particulier sur la forme. Ce descripteur est l'histogramme des coordonnées polaires relatives de tous les autres points. La détection est basée sur l'hypothèse que les points correspondants sur les deux formes seront idéalement un contexte de forme similaire. Le descripteur a été appliqué à une variété de problèmes de reconnaissance d'objets [24], avec un arrière-plan encombrant limité.

Certaines méthodes se concentrent sur la morphologie spécifique des mains et tentent de les détecter en se basant sur les caractéristiques de la forme de la main telles que le bout des doigts. Les approches dans [25] [26] utilisent la courbure comme un signal pour la détection du bout des doigts.

1.3.1.3. Détection à base de modèle 3D

Une catégorie d'approches utilise des modèles de main 3D pour la détection des mains en images. L'un des avantages de ces méthodes est qu'elles peuvent réaliser une détection de vue indépendante. Les modèles 3D employés devraient avoir suffisamment de degrés de liberté pour s'adapter aux dimensions de la main(s) présents dans une image.

Différents modèles nécessitent des fonctions d'image différentes pour construire des fonctionnalités du modèle correspondances. Les caractéristiques de point et de ligne sont utilisées dans des modèles cinématiques pour récupérer les angles formés au niveau des articulations de la main [27] [28]. En plus les postures de la main sont alors estimées à condition que les correspondances entre le modèle 3D et les caractéristiques de l'image observée soient bien établies. Dans [29], un modèle de main pleine est proposé, qui possède 27 degrés de liberté (DOF) (6 degrés de liberté pour la localisation et l'orientation 3D et 21 degrés de liberté pour l'articulation).

Dans [30], le doigt de l'utilisateur a été détecté dans les deux images d'un couple stéréoscopique calibré. Dans ces images, les deux points où cette astuce semble établir une correspondance stéréo, qui est utilisé pour estimer la position du bout du doigt est un espace 3D. À son tour, cette position est utilisée par le système pour estimer la distance du doigt sur le bureau et, par conséquent, de déterminer si l'utilisateur le touche. Dans [31], est décrit un système pour le suivi de la position et de l'orientation 3D d'un doigt à l'aide de plusieurs caméras. Le suivi est basé sur la combinaison de plusieurs sources d'information, y compris des images stéréo en distance, segmentation de la couleur et l'information de forme.

Dans [32], l'information stéréoscopique est utilisée pour fournir des positions 3D des centres de gravité de la main et des doigts, mais aussi pour reconstruire le contour des mains 3D détectées et suivies en temps réel. Dans [33] les correspondances stéréo de doigts multiples ont été utilisées pour calibrer une paire stéréo. Beaucoup d'autres approches indirectes pour la détection des doigts ont été employées, comme l'analyse d'images en utilisant spécialement des noyaux de Gabor calibrés [34]. Le principal inconvénient de l'utilisation des doigts comme des caractéristiques est qu'ils peuvent être occultés par le reste de la main. Une solution à ce problème d'occlusion implique l'utilisation de plusieurs caméras [27]. D'autres solutions sont basées sur l'estimation de la position du bout des doigts occlus, basée sur la connaissance du modèle 3D du geste en question [28] [27].

1.3.1.4. Mouvement

Une tâche fréquente dans l'analyse visuelle de scènes naturelles est le suivi d'objets en mouvement. La reconnaissance d'un objet du monde réel dans un environnement non artificiel est généralement une tâche assez complexe et a probablement besoin de plus de temps de calcul.

On peut classer les méthodes de détection de mouvement en trois catégories : les méthodes de soustraction de fond [35] [36] [37], celles qui s'appuient sur un calcul du flot optique [35] [38], et celles qui se basent sur la différence entre deux images successives [39] [40].

Le mouvement est un repère utilisé par quelques approches pour la détection des mains. La raison en est que la détection de la main basée sur le mouvement exige une configuration très contrôlée, car elle suppose que le seul mouvement dans l'image est dû au mouvement de la main. En effet, les premières œuvres (par exemple [41], [42]) supposent que le mouvement de la main est le seul mouvement produit dans l'environnement. Dans les approches les plus récentes, des informations de mouvement sont combinées avec des repères visuels supplémentaires. La différence de luminance de pixels de deux images successives est proche de zéro pour les pixels de l'arrière-plan. En choisissant et en maintenant un seuil approprié, des objets en mouvement sont détectés dans une scène statique.

Dans [43], une caractéristique nouvelle, à base de résidus de mouvement, est proposée. Généralement les mains subissent des mouvements non rigides, parce qu'ils sont des objets articulés. Par conséquent, la détection des mains capitalise sur une observation qui ne concerne que les mains, des changements d'apparence inter-frames sont plus fréquents que pour d'autres objets tels que les vêtements, le visage, et le fond.

1.3.1.5. Constatation

On a abordé dans cette partie différentes méthodes pour la détection de la main dans une scène.

La méthode basée sur la couleur de peau contient deux principales contraintes qui sont la difficulté de déduire la couleur de la main avec les changements de luminosité et ce

notamment dans un contexte mobile. La 2^{ème} contrainte est la confusion avec les objets de fond quand ces derniers ont la même distribution de couleur que la main, ce qui est très souvent le cas avec une caméra frontale d'un mobile qui rassemble main et visage dans une même scène.

La méthode basée sur la forme nécessite une analyse statique à partir d'une base de données de modèles. La première difficulté est d'alimenter cette base de modèles et la deuxième est le temps de calcul que prend la recherche de correspondances entre l'image de la caméra ainsi que les modèles de la base. Ce calcul dans un contexte mobile à 24 images par seconde en moyenne est tout simplement impossible. Une autre contrainte avec une caméra mobile est la difficulté à distinguer des formes claires mais plutôt des objets flous du fait du mouvement à la fois de la caméra et des objets filmés.

Pour la méthode 3D, la même contrainte de performance se pose qu'avec les méthodes basées sur la forme. De plus, la nécessité de disposer d'un dispositif stéréoscopique de deux caméras nous a écarté de cette méthode.

Enfin, la méthode basée sur le mouvement nécessite un arrière-plan fixe et la présence de la main en mouvement en premier plan. Le mobile étant en mouvement continu, tout mouvement indésirable et inutile est détecté en surplus.

Notre approche est de nous baser sur cette dernière méthode en l'adaptant et en l'améliorant pour éliminer les mouvements indésirables.

1.3.2. Le suivi à la trace

Le tracking (suivi) des régions segmentées de la main ou de ses caractéristiques, est la deuxième étape dans le processus de la compréhension des mouvements de la main.

Le tracking fournit l'inter-liaison de l'apparence de la main / doigt, donnant lieu à des trajectoires de ses caractéristiques dans le temps. Ces trajectoires transmettent des informations essentielles concernant le geste et peut être utilisé soit sous une forme brute en temps réel (par exemple, dans certaines applications de contrôle comme les dessins virtuels

de la trajectoire) ou après une analyse asynchrone postérieure (par exemple la reconnaissance d'un certain type de geste de la main).

1.3.2.1. Suivi basé sur le modèle

Cette classe de méthodes présente une grande similitude avec les méthodes de détection de la main. La 1^{ère} méthode utilise les caractéristiques de la main dans l'image précédente comme un modèle pour détecter la même main dans l'image suivante et ainsi déduire son nouvel emplacement, la 2^{ème} méthode quant à elle ne se base pas sur les caractéristiques au sens forme de la main mais plutôt de ses couleurs pour détecter l'objet à travers les différentes images et ainsi déduire sa position.

Dans les méthodes [44] [45], le modèle à la base de la corrélation correspondant est utilisé pour suivre à la trace des caractéristiques de la main à travers des images ou ce qu'on appelle des cadres (*frame*). Une fois la main a été détectée dans une image, la zone d'image dans laquelle la main apparaît est utilisée comme prototype pour détecter la main dans l'image suivante. Encore une fois, l'hypothèse est que les mains apparaissent dans le même quartier spatial. Cette technique est employée pour une caméra statique comme dans [46], pour obtenir des motifs caractéristiques (ou *signatures*) de gestes, à partir d'un point de vue particulier. Le travail dans [47] aborde également un éclairage variable, donc la cible est vue sous différentes conditions d'éclairage. Ensuite, on réalise un ensemble d'images de base qui peuvent être utilisées pour calculer l'apparition de l'objet vu sous des conditions d'éclairage différentes.

Concernant la 2^{ème} méthode, quelques approches détectent des mains comme des blobs d'image dans chaque cadre et font correspondre temporellement les blobs qui arrivent dans des emplacements proches à travers des cadres. Les approches qui utilisent ce type de suivi de blob sont principalement celles qui détectent des mains en se basant sur la couleur de peau, la blob étant la région d'image proportionnellement segmentée (par exemple [15]). Des approches à base de blob peuvent conserver le suivi des mains même quand il y a de grandes variations d'une image à une autre.

Étendant les approches ci-dessus, des contours déformables, ou "des serpents" ont été utilisés pour suivre à la trace des régions de la main dans des cadres d'image successifs [48]. Un serpent est un contour actif pour représenter les contours de l'objet. Les algorithmes de serpents traditionnels sont souvent utilisés pour représenter le contour d'un objet unique. Cependant, s'il y a plus d'un objet dans l'image, le modèle de serpent doit être adapté pour déterminer le contour correspondant de chaque objet.

En règle générale, la limite de cette région est déterminée par l'intensité ou du dégradé de couleurs. Néanmoins, d'autres types de caractéristiques de l'image (texture par exemple) peuvent être pris en considération. La technique est initialisée en plaçant un contour proche de la région d'intérêt. Le contour est ensuite itérativement déformé vers les bords à proximité pour mieux se répondre sur la région de la main réelle. Lorsque les serpents sont utilisés pour le suivi, un modèle de forme active est appliqué à chaque image et la convergence du serpent dans ce cadre est utilisée comme un point de départ pour le prochain cadre. Les serpents permettent le suivi en temps réel et peuvent manipuler des cibles multiples. Ils présentent de meilleures performances quand il y a suffisamment de contraste entre le fond et l'objet [49].

1.3.2.2. Techniques d'estimation optimales

Les fonctionnalités de suivi ont été largement étudiées dans la vision par ordinateur. Dans ce contexte, le cadre d'évaluation optimal fourni par le filtre de Kalman [50] a été largement utilisé dans la transformation des observations (caractéristiques de détection) dans les évaluations (la trajectoire extraite). Les raisons de sa popularité sont les performances temps réel, le traitement de l'incertitude, et la fourniture de prévisions pour les trames successives.

Plusieurs méthodes utilisent le filtre de Kalman pour prédire la localisation de la main dans une trame d'image sur la base de son emplacement détecté dans la trame précédente. Le filtre de Kalman est utilisé pour suivre la région centrée sur la main, afin d'accélérer la segmentation de la main et de choisir la région de la peau correctement lorsque plusieurs zones d'image ont la même couleur que la peau. Le filtre de Kalman suit les mouvements de la main image par image pour fournir un point de départ exact pour rechercher une région de couleur de la peau, qui est la valeur la plus proche de l'estimation. Les Résultats estimés par

le filtre sont utilisés comme point de départ pour la recherche d'une zone de couleur de peau dans la trame suivante [51].

Dans [52], l'orientation de la main de l'utilisateur a été estimée en continu avec le filtre de Kalman pour localiser le point de l'espace que l'utilisateur indique, par l'extension du bras et du pointage avec l'index. Dans [53], les mains sont suivies de plusieurs caméras, avec un filtre de Kalman dans chaque image, pour estimer les postures de la main 3D.

Dans [54], un système de suivi d'une seule personne par segmentation de la couleur de l'image en blobs utilise ensuite des informations antérieures sur la couleur de peau et la topologie du corps d'une personne pour interpréter l'ensemble de blobs comme une figure humaine. Dans [55], on propose une méthode de suivi du mouvement humain en groupant des pixels dans des blobs basées sur le mouvement cohérent, la couleur et le support temporel. Chaque blob est ensuite suivie à l'aide d'un filtre de Kalman.

1.3.2.3. Filtrage particulaire

Les filtres particulaires ont été utilisés pour suivre la position des mains et la configuration des doigts dans l'encombrement visuel dense. Dans cette approche, la croyance du système en ce qui concerne l'emplacement d'une main est modélisée par un ensemble de particules.

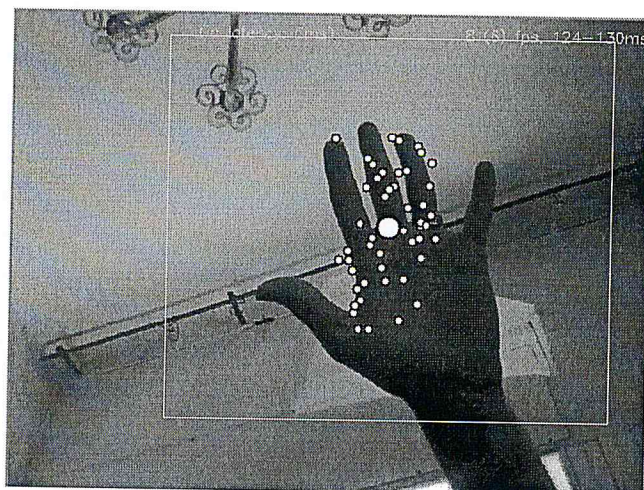


Figure 1.3 : Exemple d'application du filtre particulaire dans la détection d'une main

Un inconvénient des filtres à particules, c'est que pour les modèles complexes (comme la main de l'homme) de nombreuses particules sont nécessaires, ce qui rend le calcul très complexe en particulier pour les modèles à haute dimension. Par conséquent, d'autres hypothèses sont souvent utilisées pour réduire le nombre de particules. Par exemple dans [56], la dimensionnalité est réduite en modélisant les contraintes généralement connues en raison de l'anatomie de la main.

L'algorithme de *condensation* [56] qui a été utilisé pour apprendre à suivre des courbes sur des fonds encombrés, il présente de meilleures performances que les filtres de Kalman, et fonctionne en temps réel. La condensation utilise des modèles dynamiques appris à partir des observations visuelles. Les résultats sont très robustes pour le suivi de mouvement agile.

En général, les techniques de suivi de contour ne permettent qu'à un petit sous-ensemble de mouvements possibles pour maintenir la déformation continue des contours. Cette limitation a été surmontée dans une certaine mesure [57], qui décrit une adaptation de l'algorithme de condensation pour le suivi à travers des discontinuités dans les formes de contour.

1.3.2.4. Constatation :

Nous avons abordé dans cette partie différentes méthodes de suivi à la trace.

Le suivi basé sur le modèle qui utilise la forme ou les couleurs de la peau pour détecter l'objet n'a pas été retenu dans notre cas puisque la méthodologie choisie de notre part est basée sur le mouvement.

Concernant les techniques d'estimation optimales et le filtre particulaire, elles nécessitent un temps de calcul conséquent qui est éliminatoire dans un contexte mobile.

Dans notre cas, la main est déjà détectée dans la première étape et il nous reste à tracer sa trajectoire. Pour cela nous avons utilisé notre propre algorithme qui sera expliqué ultérieurement.

1.3.3. La reconnaissance

L'objectif général de la reconnaissance gestuelle de la main, c'est l'interprétation du mouvement de la main ou de sa posture. Fondamentalement, il y a deux types d'interactions entre la main de l'utilisateur avec un ordinateur. Le premier est celui des applications de contrôle telles que le dessin, où l'utilisateur dessine une courbe alors que l'ordinateur rend cette courbe sur une toile 2D [58] [28]. Les méthodes qui se rapportent au moteur de contrôle de la main se concentrent sur la détection et le suivi de certaines caractéristiques (par exemple, le bout du doigt, le centre de gravité de la main dans l'image). Le deuxième type d'interaction implique la reconnaissance de postures de la main, ou les signes. Naturellement, le vocabulaire des signes et gestes dépend en grande partie de l'application. En règle générale, plus le vocabulaire est large, plus la tâche de reconnaissance devient difficile.

Les deux premiers systèmes indiquent la différence entre la reconnaissance [59] et le contrôle [60]. Ainsi le premier reconnaît 25 affiches de l'alphabet international de la main, tandis que le second a été utilisé pour soutenir l'interaction dans un espace de travail virtuel.

Les arbres de décision et les réseaux de neurones sont deux techniques utilisées dans la reconnaissance du mouvement de la main et ses postures. L'étude suivante explique ces deux techniques et présente quelques travaux qui les adoptent.

1.3.3.1. Arbre de décision

Un arbre de décision peut être utilisé pour classer les cas en partant de la racine de l'arbre et en se déplaçant à travers l'arbre jusqu'à ce qu'une feuille soit rencontrée. A chaque nœud de décision qui n'est pas une feuille, le résultat de cas pour le test au niveau du nœud est déterminé et le pointeur se déplace vers la racine du sous arbre correspondant à ce résultat. Lorsque ce processus aboutit finalement à une feuille, la classe de notre cas est prévue pour être celle enregistrée sur la feuille. [61]

Dans le travail de *Yiqiang et al* [61], on se base sur les arbres de décision pour la reconnaissance de postures de la main. En utilisant deux *CyberGlove* (gant de données) avec 18 capteurs et un *Traquer* avec deux récepteurs placés sur le poignet de chaque *CyberGlove*, soixante-cinq différents gestes de la main sont classés comme résultat.

Toujours dans le contexte de la reconnaissance de postures de la main, L'équipe J. Mackie et B. McCane [62] ont démontré un système de classification d'arbres de décision formés sur des données synthétiques capables de classer des images de vraies mains. Dans leurs résultats, La détection des doigts avec des arbres de décision prévoit un système de classification rapide approprié pour gestes de la main et Le système était capable de détecter la présence ou l'absence de chacun des doigts dans une image de la main. Cela peut être utilisé comme information principale, pour devenir des identifiants pour des gestes. Le temps de formation est de quelques secondes pour chaque arbre, ce qui le rend acceptable au démarrage. Ceci permet aux différents ensembles de données d'être chargés au démarrage.

1.3.3.2. Les réseaux de neurones (ANN):

Les réseaux de neurones sont constitués d'éléments simples fonctionnant en parallèle. Ces éléments sont inspirés par les systèmes nerveux biologiques. Comme dans la nature, la fonction de réseau est déterminée en grande partie par les liens entre les éléments. On peut former un réseau de neurones pour réaliser une fonction particulière en réglant les valeurs des connexions (poids) entre les éléments. Les réseaux de neurones sont ajustés ou formés, de sorte qu'une entrée particulière conduit à une sortie cible spécifique. [63]

Un réseau neuronal artificiel est un système adaptatif qui modifie sa structure basée sur des informations externes ou internes et qui coule à travers le réseau au cours de la phase d'apprentissage. L'utilité de modèles de réseaux neuronaux artificiels réside dans le fait qu'elles peuvent être utilisées pour déduire une fonction de l'observation. Ceci est particulièrement utile dans les applications où la complexité des données ou des groupes rend la conception d'une telle fonction à la main impraticable. Les tâches auxquelles les réseaux de neurones artificiels sont appliqués sont la classification, y compris les motifs et la reconnaissance de la séquence ainsi que la détection de la nouveauté et de la prise de décision séquentielle. [64]

Dans le travail de Noor A. Ibraheem et Rafiqul Z. Khan [65], la reconnaissance de gestes de la main, qui est basée sur la distribution gaussienne des caractéristiques des doigts, prend en compte la possibilité d'un doigt appartenant à l'une des cinq catégories possibles de même que la probabilité de toutes les combinaisons de doigts possibles pour correspondre à

un geste de la main d'entrée. On trouve à partir de ces expériences que le taux de reconnaissance est très prometteur et se rapproche de 90,45%.

1.3.3.3. Constatation

Nous avons étudié dans cette partie différentes approches pour la reconnaissance des gestes et postures. Parmi ces approches, l'arbre de décision et réseau de neurone sont adoptés dans quelques travaux. En revanche, ces deux approches nécessitent une base peuplée d'une grande quantité de modèles un donc temps d'initialisation conséquent ce qui les rend moins pratique pour les prototypes d'interface utilisateur qui utilisent de simples gestes.

Nous avons adapté dans notre approche un autre algorithme pour la reconnaissance des gestes, nommé *\$I Recognizer*, qui est basé sur la comparaison géométrique, cet algorithme est utilisé pour l'apprentissage automatique à base d'exemples. Le prochain chapitre l'explique en détail et met en avant nos améliorations.

1.3.4. Conclusion

D'après l'étude faite dans ce chapitre, nous avons pu constater qu'il y eu de nombreuses recherches et tentatives d'approche en terme de reconnaissance gestuelle, afin de l'incorporer dans la vie quotidienne.

Puisque notre approche est divisée en 3 étapes : détection, suivi à la trace et reconnaissance, la seconde partie de ce chapitre les retranscrit dans cet ordre. Dans chacune de ces étapes, on a étudié quelques techniques existantes, afin de s'inspiré de certaines d'entre elles tout en y apportant nos propres améliorations.

Pour la première étape, nous avons choisi la technique de détection basée sur le mouvement mais nous nous sommes appliqués à l'améliorer pour éliminer tous les mouvements inutiles et indésirables.

La deuxième partie qui est le suivi à la trace, on a décidé de développer notre propre algorithme.

La troisième étape de reconnaissance, nous avons implémenté un algorithme déjà existant (l'algorithme \$1 recognize), on le modelant de façon à correspondre au contexte mobile

CHAPITRE 2 : APPRENTISSAGE AUTOMATIQUE :

Apprentissage à partir d'exemple

2.1. Introduction

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains traits descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. Les méthodes utilisées par les systèmes d'apprentissage sont très nombreuses et sont issues de domaines scientifiques variés.

L'apprentissage automatique, dans une définition très générale, consiste en l'élaboration de programmes qui s'améliorent avec l'expérience. Les applications sont nombreuses et concernent des domaines très variés. On peut citer, par exemple, la reconnaissance de formes avec, en particulier, la reconnaissance de la parole et du texte écrit, le contrôle de processus et le diagnostic de pannes, les programmes de jeu. Les méthodes d'apprentissage à partir d'exemples sont très utilisées dans la recherche d'informations dans de grands ensembles de données. En effet, l'évolution de l'informatique permet de nos jours de manipuler des ensembles de données de très grande taille (« datawarehouse » ou « entrepôt de données »). [66]

2.2. Apprentissage à partir d'exemples : l'algorithme « \$1 Recognizer »

Pour faciliter l'intégration des gestes dans des prototypes d'interface utilisateur, nous présentons un système de reconnaissance « \$1 recognizer ». Dans ce système la reconnaissance est très simple, impliquant seulement la géométrie de base et la trigonométrie. Il prend en charge la rotation configurable, l'échelle et la position invariance, n'exige pas la sélection de fonction ou des exemples de formation, est résistant aux variations de l'échantillonnage d'entrée et prend en charge des taux de reconnaissance élevés, même après seulement un exemple représentatif. Bien que le \$1 a ses limites en raison de sa simplicité, il offre un excellent taux de reconnaissance pour les types de symboles qui peuvent être utiles dans les interfaces utilisateur. [67]

2.2.1. Définition

Le \$1 recognizer est un simple algorithme de reconnaissance de geste créée par Andy Wilson de Microsoft Research et Jacob Wobbrock et Yang Li de l'Université de Washington. Conçu pour le prototypage rapide d'interfaces utilisateur fondées sur des gestes. En termes d'apprentissage automatique, \$1 est un exemple à base de classificateur du plus proche voisin avec une fonction de notation euclidienne, c'est à dire, un comparateur de modèle géométrique. Malgré sa simplicité, le \$1 ne nécessite que très peu de modèles pour être bien performant et seulement d'environ 100 lignes de code, ce qu'il le rend facile à déployer. Le \$1 est distribuée sous le nouvel accord de licence BSD. [68]

2.2.2. Caractéristiques

Les résultats d'un geste utilisateur ce résume par un ensemble de points appelé « Candidat 'C' », et nous devons déterminer quel ensemble de points de modèle précédemment enregistrées « Template 'Ti' » elle se rapproche le plus.

Les Candidats et le modèle de points sont habituellement obtenus par des moyens interactifs, en utilisant des chemins de décisions, dans la région de détection.

2.2.3. Algorithme

Les gestes destinés à servir de modèles, ou les gestes candidats qui tentent d'être reconnus, sont d'abord traités de la même façon c'est-à-dire: ils sont redimensionnés, tournés une fois, mis à l'échelle, et traduits. Les points candidats C sont ensuite comparé avec chaque ensemble de points modèle Ti sur une série de réglages angulaires de C afin de trouver son alignement angulaire optimal à Ti.

L'Algorithme « \$1 recognizer » fonctionne en utilisant un processus de 4 étapes.

- **Étape 1 : Échantillonner le chemin de point**

Les gestes dans les interfaces utilisateur sont différents. Ainsi, la vitesse de mouvement aura un effet évident sur le nombre de points d'entrée.

La première étape de l'algorithme consiste à échantillonner le trajet enregistré dans un nombre fixe de points qui sont régulièrement espacées le long du trajet.

La figure 2.1 représente un point d'interrogation et un triangle lent et rapide faite par l'utilisateur utilisant un stylet sur un Pocket PC. On note les différences de temps considérables et des chiffres résultant de points.

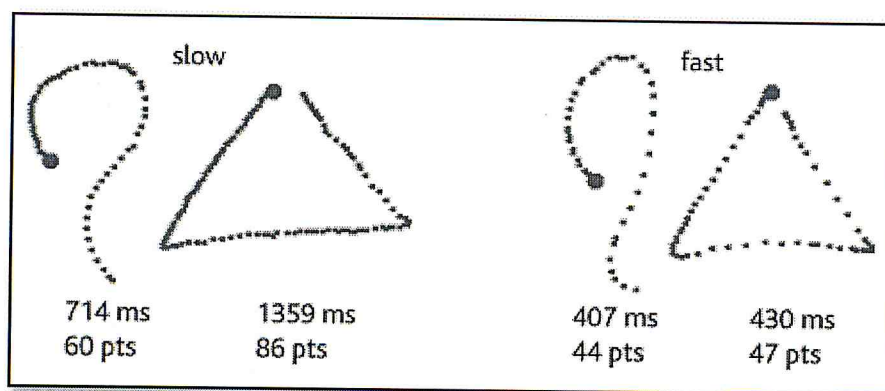


Figure 2.1 : différences de temps considérables et des chiffres résultant de points.

N.B. : On note les différences de temps considérables et des chiffres résultant de points.

Pour faire des chemins de geste directement comparables, même à des vitesses différentes de mouvement, l'algorithme échantillonne les gestes tels que le chemin défini par leurs points d'origine M est définie par N points équidistants (figure 2.2). L'utilisation d'un N qui est trop faible entraîne une perte de précision, tout en utilisant un N qui est trop élevé ajoute le temps de comparaisons de chemin. En règle générale le N est borné par $32 \leq N \leq 256$.

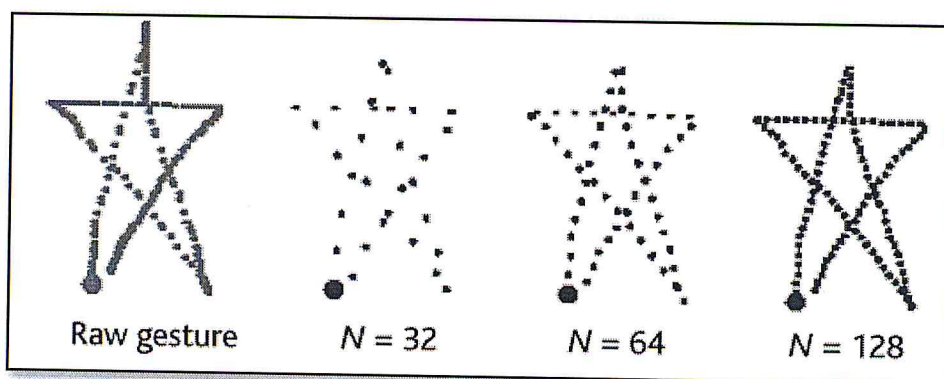


Figure 2.2 : Un geste « étoile » échantillonné à $N = 32, 64$ et 128 points.

- **Etape 2 : Rotation basé sur «Indicated Angle »**

Tout d'abord, l'algorithme trouve l'angle indicative «Indicated Angle » d'un geste, et le défini comme l'angle formé entre le centre de gravité du geste et le premier point du geste. Ensuite, il fait tourner le geste afin que cet angle soit à 0° (figure 2.3).

La rotation se fait de telle sorte que le premier point est directement à droite du centre de gravité.

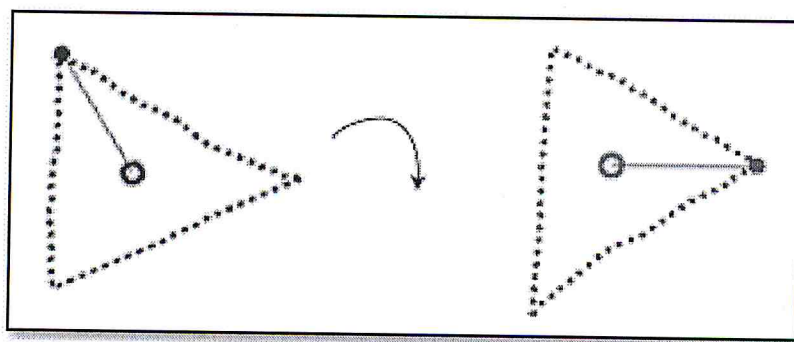


Figure 2.3 : Rotation d'un triangle de sorte que son «angle indicative» est à 0° .

- **Etape 3 : Mettre à l'échelle**

Cette phase met à l'échelle la trajectoire à une hauteur et une largeur fixes, et prend en considération la distance des points du candidat C.

- **Etape 4 : Trouver l'angle optimal pour le meilleur score**

À ce stade, tous les candidats et les modèles ont été traités de la même façon. Ensuite, l'algorithme applique l'étape 4, qui fait réellement la reconnaissance : Pour chaque trajectoire de référence, on calcul la distance moyenne des points correspondants dans le chemin d'entrée. Le chemin qui a la plus faible moyenne de distance de point est le correspondant.

2.2.4. Limitation de l'algorithme \$1 :

Pour faciliter les comparaisons par paires de points, l'algorithme \$1 par défaut fait la rotation, l'échelle et la position invariable. Bien que cela offre une tolérance aux variations de geste, cela signifie que \$1 ne peut pas distinguer les gestes dont l'identité dépendra des orientations spécifiques. En outre, les lignes horizontales et verticales sont traduites par un même modèle, et on ne peut pas faire une différence entre eux.

Enfin, \$1 n'utilise pas l'heure (time), donc les gestes ne peuvent être différenciées en fonction de la vitesse.

2.2.5. Améliorations adaptés.

Nous avons utilisé l'algorithme \$1 dans notre approche pendant la phase de la reconnaissance, la raison de notre choix correspond à notre cas d'étude. Afin d'adapter cet algorithme dans un cadre de reconnaissance gestuelle sur mobile, nous l'avons amélioré pour le rendre fonctionnel dans notre cas. Vu que l'invariance de rotation est indésirable, la modification consiste à ignorer l'étape de rotation, techniquement les candidats C et les modèles T_i présents dans la base peuvent être comparés sans faire tourner l'angle indicatif à 0° .

2.3. Conclusion

Pour un environnement mobile, où les performances sont limitées (rapidité de calcul, espace mémoire, temps de traitement, ...), l'algorithme \$1 a été un choix correspondant à notre cas, vu qu'il est utilisé pour la reconnaissance des gestes simplifiés, et surtout son temps de réponse rapide lors de la comparaison avec la base à modèles.

Côté base à modèles, l'algorithme *\$1 recognizer* traite les modèles sous forme de points dans un espace euclidien, qui ne nécessite pas un grand espace mémoire, et cet avantage qui nous a bien facilité la reconnaissance dans notre approche, puisque il fournit des réponses un temps minime. Aussi il ne nécessite pas une grande base d'apprentissage, un seul modèle par cas est suffisant.

CHAPITRE 3 : CONCEPTION

3.2. Introduction

De façon générale, notre approche utilise la camera frontale du mobile pour capturer le geste de la main fait par l'utilisateur dans le but de donner un sens à ce geste et ainsi lui associer une action.

Notre manière d'aborder la conception a été de la décomposer en sous problématiques qui simplifient l'atteinte de notre objectif principal : la reconnaissance gestuelle dans un environnement mobile.

Les différentes étapes sont la détection, le suivi à la trace et enfin la reconnaissance qui sont expliquées dans le schéma ci-dessous (Figure 3.1) :

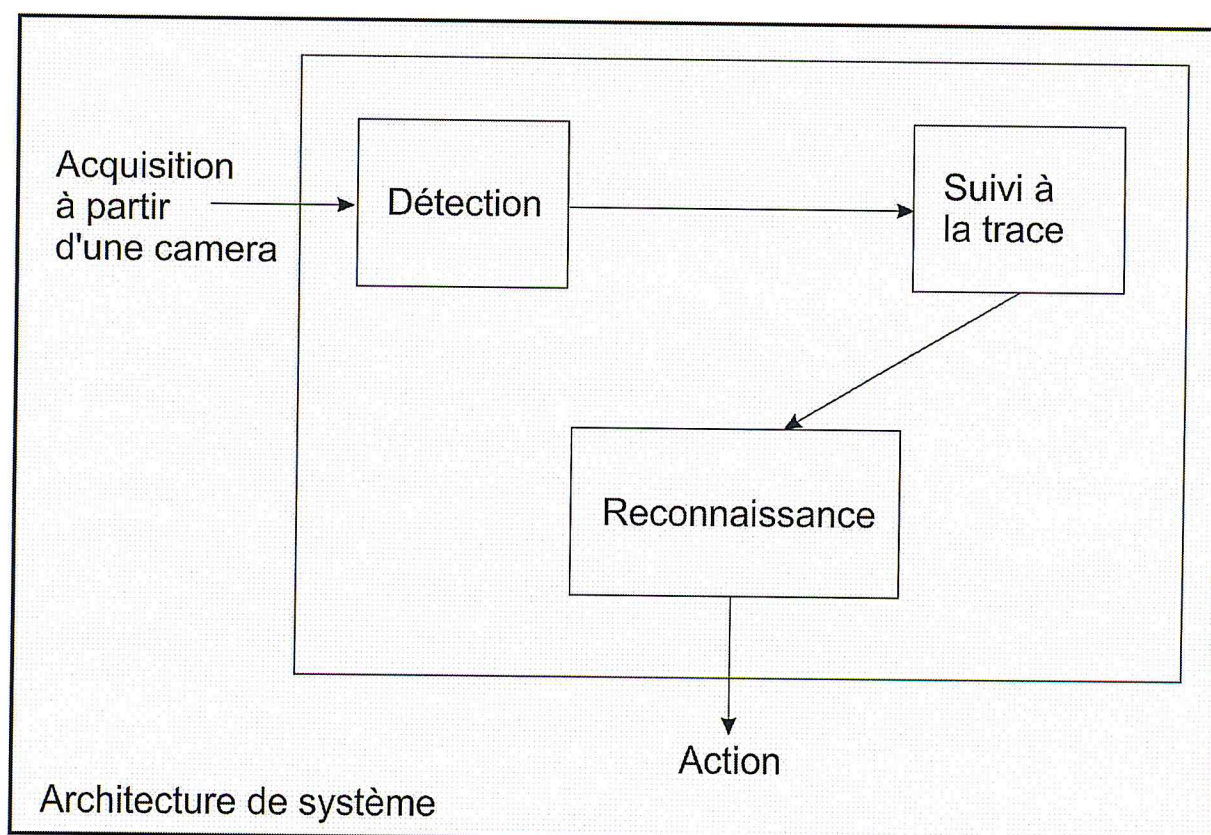


Figure 3.1 : Architecture du système.

Premièrement, la détection consistait à déterminer l'objet à détecter et dans notre cas, c'est la main de l'utilisateur. Puisque nous sommes dans un contexte mobile, nous avons à prendre en considération toutes les contraintes liées à cela.

Deuxièmement, pour le suivi à la trace, nous avons choisi une méthode pour tracer le mouvement de l'objet grâce à son centre de gravité tout au long de la durée du geste.

Et enfin, l'étape de reconnaissance de gestes qui consiste à faire correspondre à un mouvement particulier, une action particulière.

On peut expliquer le schéma ci-dessus par un pseudo-algorithme de tout le processus du système, et on verra en détail les différentes étapes :

Début

Définir la valeur de **CONTRASTE**.

Définir la valeur de **FLOU**.

Définir la valeur de **CHANCES**.

Acquérir une image **IMG_A** à partir de la camera.

TANT QUE *VRAI*

Faire

// Détection

Acquérir une image **IMG_B** à partir de la camera.

Faire la différence entre **IMG_A** et **IMG_B** et mettre le résultat dans **IMG_D**.

Augmenter le contraste de l'image **IMG_D** avec une valeur **CONTRASTE**.

Appliquer le filtre flou à l'image **IMG_D** avec une valeur **FLOU**.

Encadrer les pixels blancs de l'image **IMG_D** dans une variable **ZONE**.

Mettre la valeur de l'image **IMG_B** dans **IMG_A**.

//Suivi a la trace

SI

ZONE a une surface suffisante pour dire qu'on a un mouvement brusque dans la scène

Alors SI

On a un début de mouvement

Alors

Initialiser une variable trajectoire, qui est une série de points qui représente les centres de gravité de l'objet détecté.

Fin SI

```

On initialise la valeur de chances utilisées (CHANCES_UTILISEES) à zéro.
On considère l'état actuel en mouvement.
On ajoute le centre de gravité de ZONE à la trajectoire (TRAJECTOIRE).
SINON
    SI
        ZONE n'été pas encadrée ou n'a pas une valeur suffisante ET l'état actuel est
        en mouvement
        Alors    SI CHANCES_UTILISEES < CHANCES permis
                Alors
                    Augmenter la valeur CHANCES_UTILISEES.
    SINON
        Marquer la fin de mouvement.
        Comparer la trajectoire avec la base de modelés en utilisant
        l'algorithme $1.
        Le modèle résultant sera utilisé pour associer une action
        correspondante au geste.
    Fin SI
Fin SI
Fin SI
Fin TANT QUE

```

3.3. Détection

Comme nous l'avons vu dans l'état de l'art, plusieurs techniques ont été implémentées pour détecter une main dans une scène dont celles basées sur la détection de la couleur de peau. Sachant que l'utilisateur utilise la caméra frontale de son mobile, on imagine rapidement la confusion qu'il peut y avoir entre un visage et une main dans le champ de la caméra en plus d'autres contraintes liées à l'environnement très changeant du fait que la caméra soit mobile. Une deuxième technique étudiée est la détection de la forme que nous avons rejetée puisque cette technique est très gourmande pour un processeur mobile. Nous avons choisi finalement la technique de détection basée sur le mouvement mais nous

nous sommes appliqués à l'améliorer pour éliminer tous les mouvements inutiles et indésirables.

Acquisition à partir d'une camera

Cette étape se concentre sur la récolte d'images à partir de la caméra frontale du dispositif mobile, notre système considère la scène filmée comme un ensemble d'images.

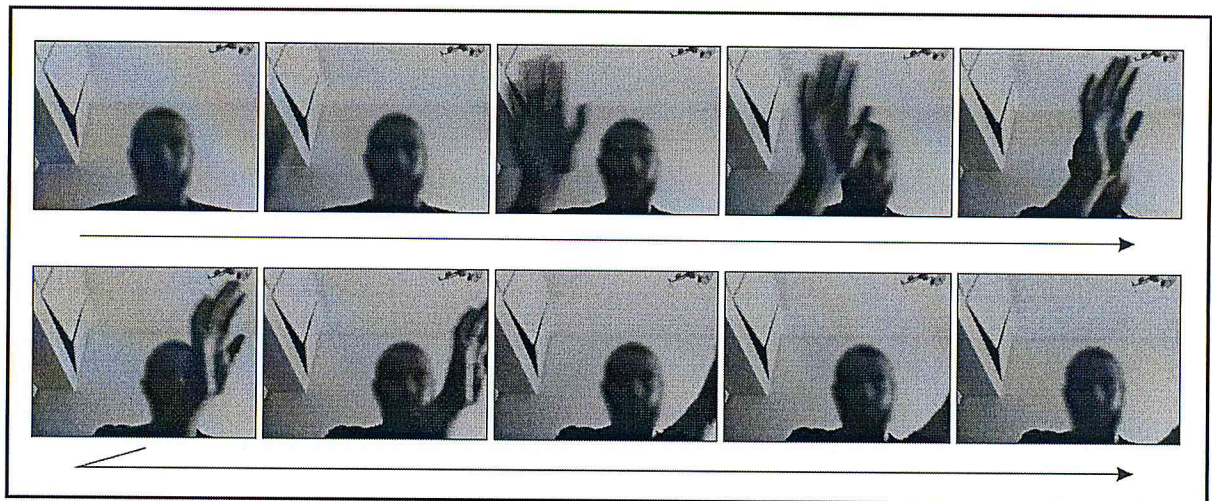


Figure 3.2 : Flux d'image acquis à partir de la caméra.

Différence

A ce niveau, on traite chaque couple d'images reçus dans les images capturées. Plus précisément, on compare chaque image avec l'image qui la précède. Le but de cette opération est de détecter le mouvement puisqu'une différence signifie qu'il y ait eu un changement dans la scène.

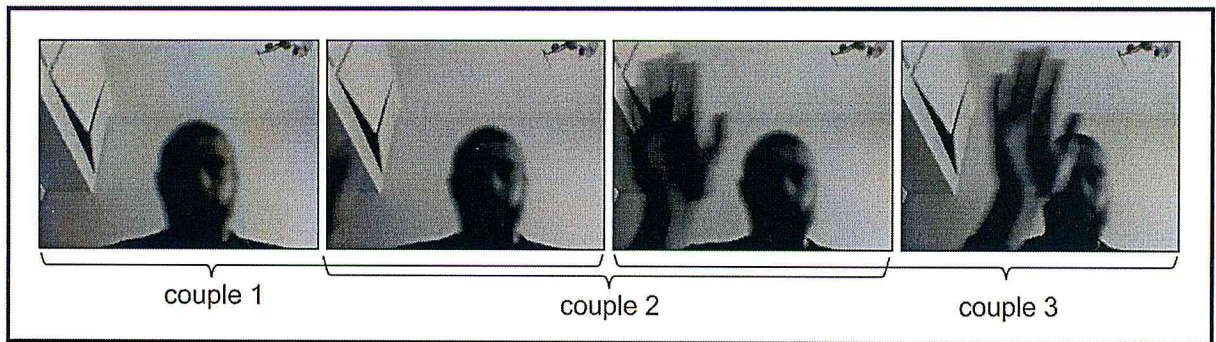


Figure 3.3 : Couplage d'images.

Pour cela, on utilise la méthode « DIFFERENCE ». Cette fonction compare les couleurs élémentaires des deux images superposées pixel par pixel.

Plus la différence est grande entre deux points des deux images, et plus le point résultant tendra vers la couleur blanche. Dans l'autre cas, plus les points sont similaires et plus la couleur résultante tendra vers le noir.

Techniquement, une soustraction est effectuée entre les couleurs des pixels comparées pour obtenir le pixel de différence.

```

POUR i ALLANT_DE 0 A HAUTEUR_IMG
  POUR j ALLANT_DE 0 A LARGEUR_IMG
     $IMG\_D[i,j] = IMG\_A[i,j] - IMG\_B[i,j];$ 
  Fin POUR
Fin POUR

```

On le voit bien sur la figure 3.4 ci-dessous qui est le résultat de la différence du 2eme couple de la figure 3.3, tous les points noirs veulent dire qu'à cet endroit de l'image, il n'y ait peu de changement.

Sur la zone à gauche où on voit la présence de la main, les zones sont plus claires puisque la différence existe à cet endroit-là.

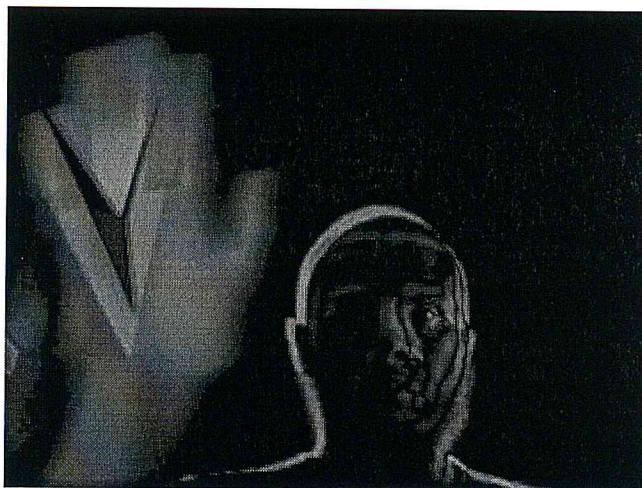


Figure 3.4 : Application d'une différence entre deux images.

Elimination des bruits

Etant donné que tout mouvement (signifiant ou insignifiant) a été détecté, cette étape consiste à éliminer les mouvements insignifiants. On suppose que tout mouvement brusque désigne la main.

Afin de réaliser ce travail, on passe par deux étapes principales : Augmentation de contraste et application du filtre flou.

Augmenter le contraste

Comme mentionné plus haut, la différence entre les images retourne une image qui contient des pixels plus ou moins claires selon le degré de différence entre chaque pixel des deux images comparées. Ces pixels claires doivent atteindre la couleur blanche, pour l'encadrer après comme l'objet en mouvement.

```

CONTRASTE = 15;
POUR i ALLANT_DE 0 A HAUTEUR_IMG
    POUR j ALLANT_DE 0 A LARGEUR_IMG
        IMG_D [i, j] = IMG_D [i, j] * CONTRASTE;
    Fin POUR
Fin POUR
  
```


On aura comme résultat cette image ci-dessous avec les changements plus visibles identifiables par leurs couleurs blanches

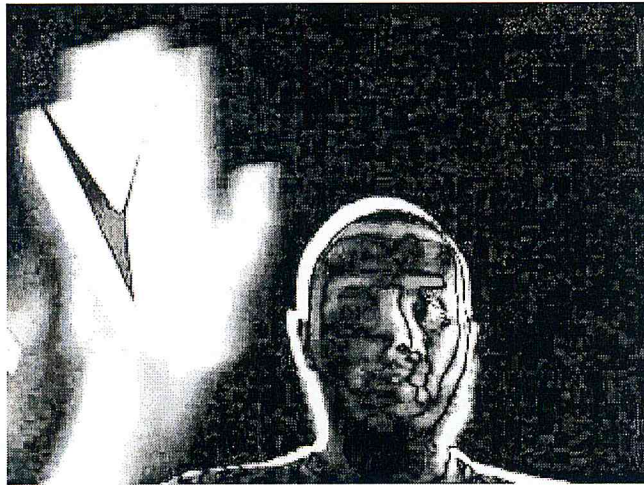


Figure 3.5 : Augmentation de contraste d'une image.

Appliquer le filtre « flou »

Cette étape consiste à cacher les zone de mouvement indésirable, le phénomène du filtre flou est que chaque pixel qui se propage sera « mixé » avec les pixels environnants de façon à ce que les petites différences, donc les contours avec des pixels blancs, soient influencées par le voisinage et perdrons ainsi leurs couleurs et deviennent plus foncées, tandis que les zones intéressantes, les surfaces blanches, qui symbolisent le mouvement de la main gardent leur couleur originale.


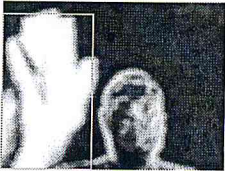
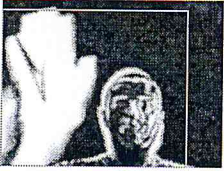
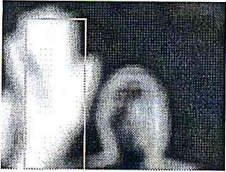
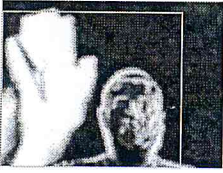
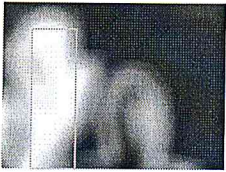
L'algorithme suivant explique comment appliquer le filtre « flou » sur une image, cet algorithme nécessite une valeur prédéfini de flou, donc nous allons utiliser comme tests les valeurs {3, 5, 7, 9, 11, 13} et de comparer les résultats après.

```

POUR ALLANT_DE A
    POUR ALLANT_DE A
        Fin POUR
    Fin POUR
Fin POUR

```

Afin de choisir la valeur du flou à appliquer, nous avons développé un programme de tests pour essayer les différentes valeurs. Dans le tableau ci-dessous nous allons voir les différents résultats des images avec la valeur du flou associée.

Valeur du flou	Image résultante	Valeur du flou	Image résultante
3		9	
5		11	
7		13	

D'après ce dernier tableau on remarque que si la valeur est inférieure à 9 on capte presque tous les mouvements, c'est-à-dire qu'on capture même les mouvements indésirables. Aussi si la valeur est supérieure à 9, on perd les contours de la main de l'utilisateur, et on risque de tomber dans des erreurs. Donc pour un choix optimal nous avons choisi bien évidemment la valeur « 9 » pour le filtre flou.

Encadrer l'objet en mouvement

Cette dernière étape va encadrer notre objet, Après l'élimination des bruits, on suppose que l'image résultante ne comporte que la main de l'utilisateur, identifiables par la couleur blanche et notre système va calculer cette surface et déduire la zone de délimitation qui sera la donnée d'entrée pour la prochaine phase, c'est à dire le suivi à la trace.

Dans le cas contraire et s'il y a une absence de zone de mouvement, ou sa surface n'atteint pas un certain seuil, on considère qu'aucun objet n'a été détecté.

```

X = HAUTEUR_IMG ; Y = LARGEUR_IMG ; H = 0 ; L = 0
BLANC = 16777215 //la valeur de numérique de blanc
POUR i ALLANT_DE 0 A HAUTEUR_IMG
    POUR j ALLANT_DE 0 A LARGEUR_IMG
        SI IMG_D [i, j] == BLANC
            Alors
                SI j < X
                    Alors X = j ;
                SINON L = j - X ;
            Fin SI
        SI i < Y
            Alors Y = i ;
            SINON H = i - Y ;
        Fin SI
    Fin SI
Fin POUR
Fin POUR

```


On voit dans la figure 3.7 ci-dessous le cadre qui a été dessiné après la détection de la main. On remarque que les contours ne sont pas pris en charge, mais que les surfaces sont bien encadrées.

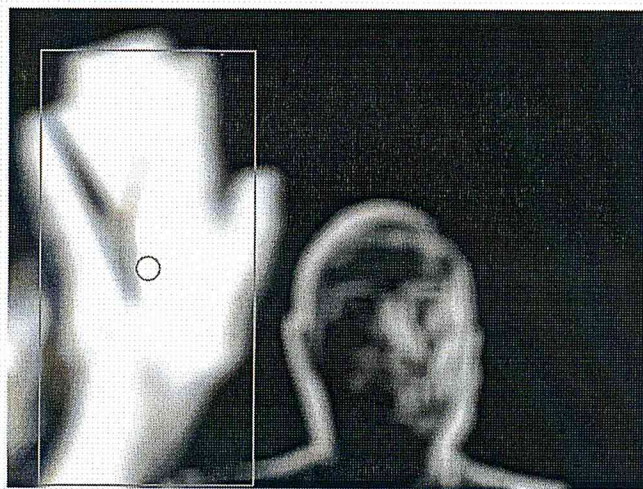


Figure 3.7 : Zone de délimitation encadrée.

3.3.1. Exemple

On voit dans la figure 3.8 ci-dessous le cadre dessiné à travers les différentes images lors du mouvement de la main. Cet exemple est le résultat obtenu dans l'étape de détection à partir des images de la figure 3.2.

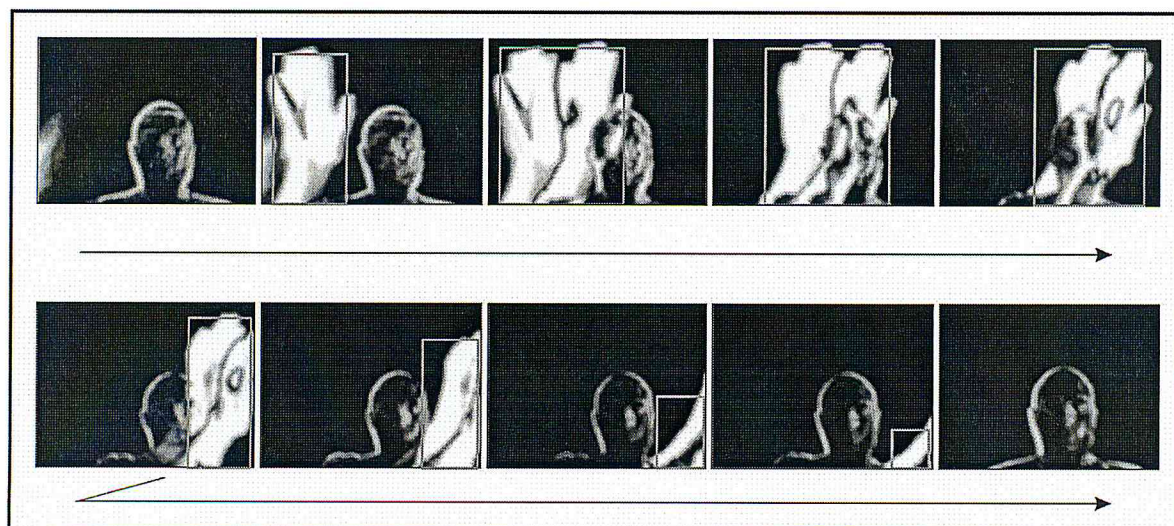


Figure 3.8 : Détection de la main dans un geste complet.

3.4. Suivi à la trace (ou tracking)

Cette étape fournit l'inter-liaison de l'apparence de la main qui bouge dans une scène donnant lieu à des trajectoires de ses caractéristiques dans le temps. Ces trajectoires transmettent des informations essentielles concernant le geste.

Ainsi, on trace une ligne virtuelle reliée par tous les centres de gravité de la main de chaque trame enregistrée par la caméra.

Une problématique rencontrée est de savoir quand le mouvement commence et quand il prend fin. Pour ça, On a suivi le déroulement suivant :

Détecter le début de mouvement

Le début du mouvement représente le moment où l'objet est détecté. (Voir la figure 3.7, 2eme trame)

Suivi du l'objet en mouvement

Comme évoqué dans l'étape de détection, l'objet en mouvement est encadré par une zone de délimitation. A cette étape, on prend le centre de chaque zone de délimitation qu'on appelle le centre de gravité. Ce processus est exécuté après la détection d'un objet dans la scène. Ainsi, tout au long du déplacement de l'objet en scène, des points successifs vont être sauvegardés et qui formeront finalement une ligne virtuelle plus tard.

Dans la figure 3.9, les points marqués en rouge sont les centres de gravité.

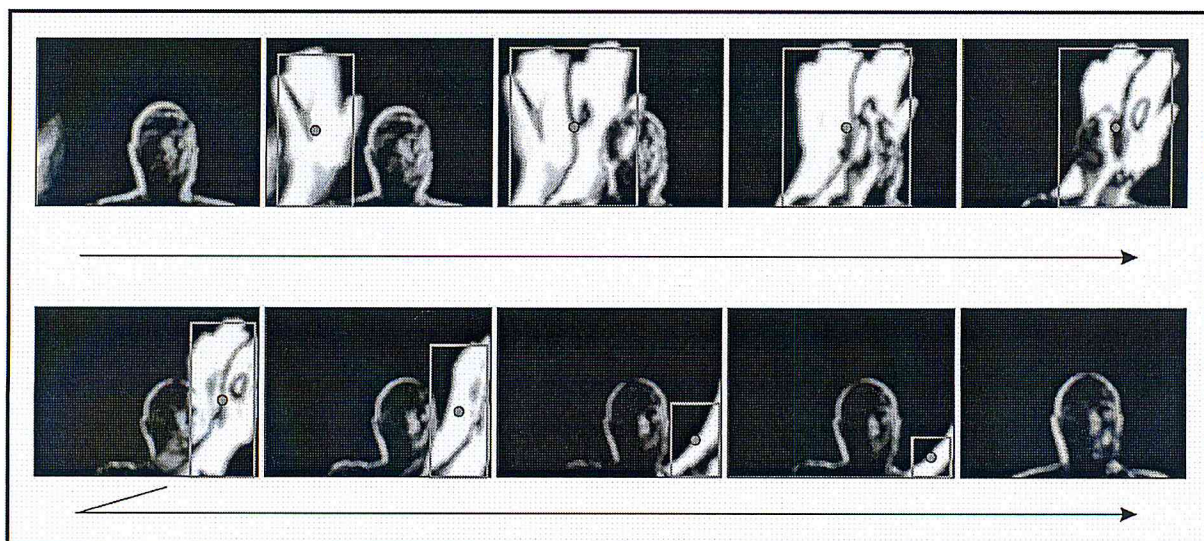


Figure 3.9 : Suivi à la trace d'un geste complet.

Détection de la fin de mouvement et dessin de la trajectoire

Cette étape coïncide avec la disparition de l'objet de la scène, à ce moment-là, tous les points sauvegardés dans l'étape précédente vont être reliés et dessineront la ligne qui représente la trajectoire parcourue par l'objet.

La figure 3.10 représente la trajectoire résultante obtenue après le suivi. Dans la scène présente sur la figure 3.9, on remarque le début du mouvement dans la trame 2, et la fin dans la trame 10, les points sauvegardés vont être reliés et dessineront la ligne qui représente la trajectoire parcourue par l'objet.

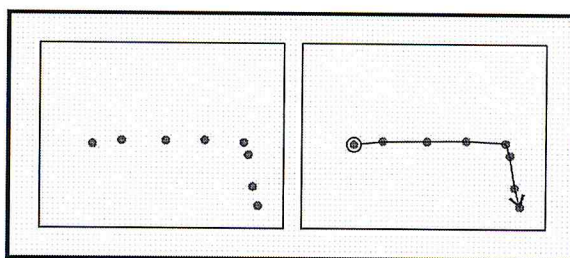


Figure 3.10 : Trajectoire parcourue par une main.

Un exemple qui schématise le déroulement expliqué :

1^{er} cas :

Dans la figure 3.11 ci-dessous, on fait un suivi de la trajectoire résultante du geste pas à pas, du début de mouvement (2), jusqu'à sa fin (6) qui coïncide avec la disparition de l'objet.

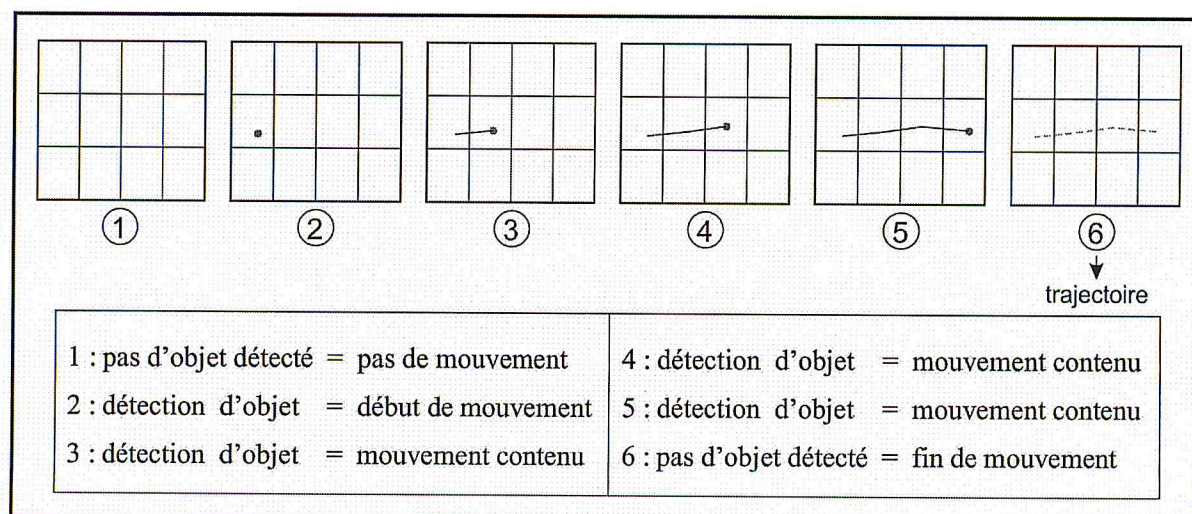


Figure 3.11 : Un exemple qui schématise un suivi dans un cas normal.

2eme cas :

Etant donné que la disparition d'objet signifie la fin du suivi, on remarque dans la figure 3.12 une interruption du suivi (4), due à la disparition inattendue de notre objet causé par la caméra, malgré la persistance du mouvement.

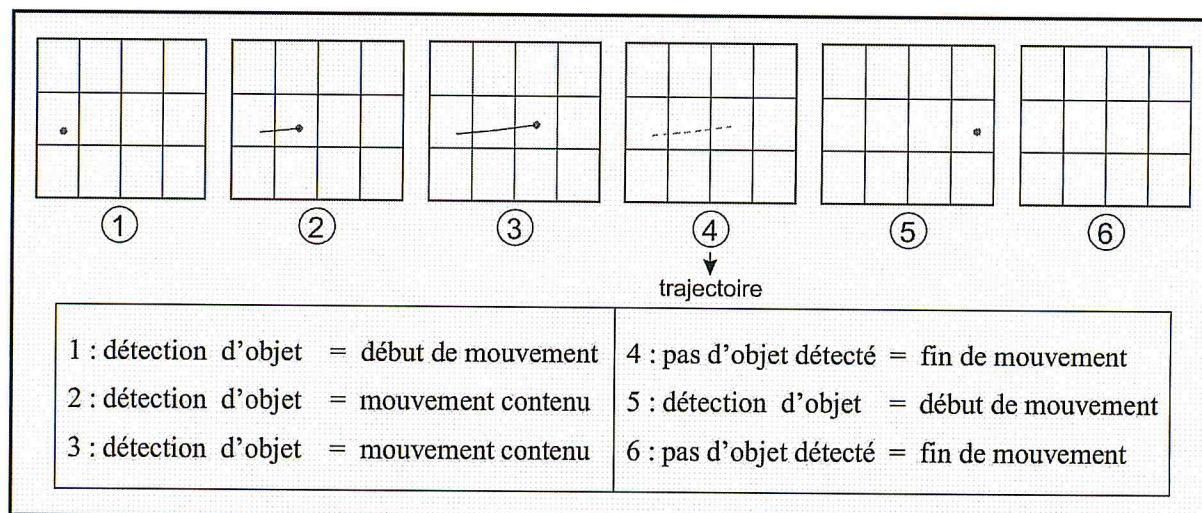


Figure 3.12 : Un exemple qui schématise un suivi incomplet.

3.4.1. La méthode « Second chance »

Un problème rencontré lors du suivi de l'objet et qui est lié à la caméra est le fait de perdre de vue l'objet, dû à un taux d'erreurs inhérents à la qualité de la caméra. De ce fait, on croit que le mouvement est terminé alors que ce n'est pas toujours le cas.

Nous avons pu résoudre cette problématique en donnant « une chance » au processus de suivi. Concrètement, une fois l'objet disparu, on attend de voir sur l'image suivante pour voir si cet objet revient sur scène et ainsi continuer à former les différents points qui dessineront notre ligne virtuelle. En cas de disparition de l'objet dans deux trames successives ou plus, un nombre de chance prédéterminer peut être donné.

Exemple :

Dans la figure 3.13 on applique la méthode « second chance » avec un nombre de chances égale à 1.

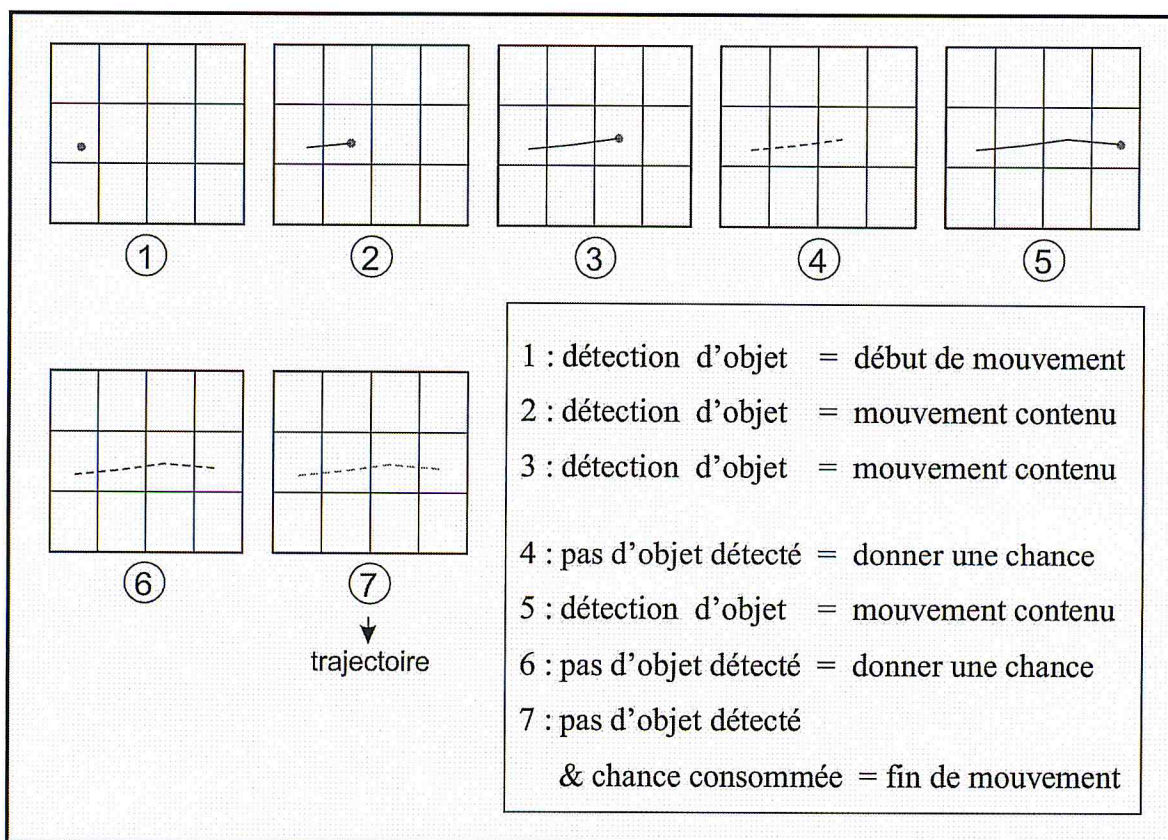


Figure 3.13 : Un exemple qui schématise la méthode « second chance ».

3.5. Reconnaissance

La meilleure façon de faire ça a été d'avoir un apprentissage automatique dans le processus de reconnaissance. Pour cela, nous avons travaillé sur l'algorithme « \$1 ».

Une fois que nous récupérons la trajectoire parcourue par la main de l'utilisateur, il nous reste à définir et à faire correspondre cette trajectoire avec le geste qui lui ressemble, et cela en comparant avec les modèles déjà classifiés dans la base de modèles.

Une base de modèles est un ensemble de données qui représentent les différents comportements des gestes et qui leur attribuent un sens approprié.

L'algorithme "\$1" nous a beaucoup aidé pendant l'implémentation de la base de modèles toute en lui appliquant quelques améliorations et de le rendre compatible avec notre système.

Construction de la base de modèles

Cette étape de reconnaissance se base sur la méthode d'apprentissage automatique basée sur les modèles, ces derniers sont ajoutés après leur avoir attribué un sens à chacun.

Reconnaissance de modèle avec L'algorithme « \$1 » amélioré

Comme expliqué ci-dessus, cette étape de reconnaissance est fondée sur l'algorithme "\$1" qui compare la trajectoire d'entrée avec les modèles disponible dans la base, en passant par plusieurs traitements.

Le système de reconnaissance 1 \$ est un comparateur de modèle géométrique, ce qui signifie que les trajectoires sont comparées aux modèles précédemment mémorisées (lors de l'initialisation du système), et le résultat obtenu est le plus proche dans l'espace euclidien.

Exemple



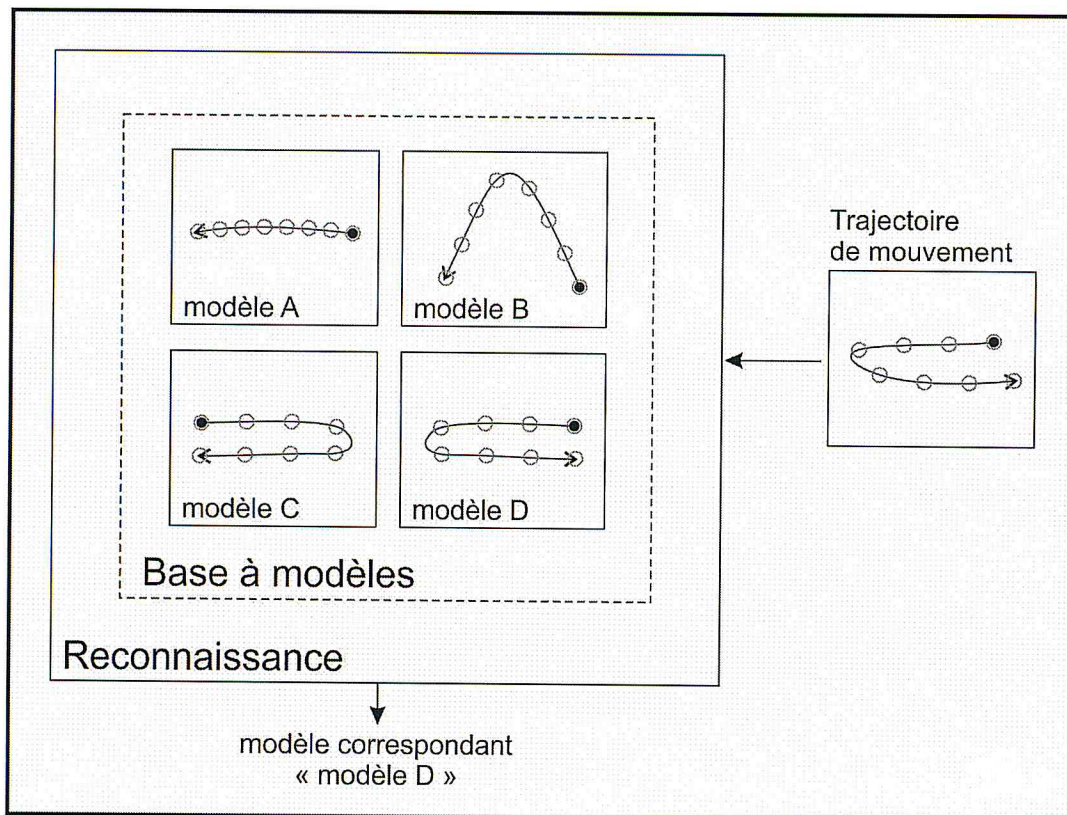


Figure 3.14 : Reconnaissance d'une trajectoire de mouvement.

Après avoir terminé la reconnaissance on déclenche l'évènement (Action) associé au geste.

```
// Reconnaissance
N = nombre de points échantillonné
C = la trajectoire de la main de l'utilisateur
T = tous les modèles de la base
TAILLE = taille de la base à modèle
POUR i ALLANT_DE 0 A TAILLE

$$D[i] = \frac{\sum_{K=1}^N \sqrt{(C[K]x - Ti[K]x)^2 + (C[K]y - Ti[K]y)^2}}{N}$$

//Di représente la distance entre C et Ti
FIN POUR

Le minimal des Di est choisi comment la distance la plus petite,
Donc on associé à la trajectoire C le modèle Ti
```


3.6. Exemple complet

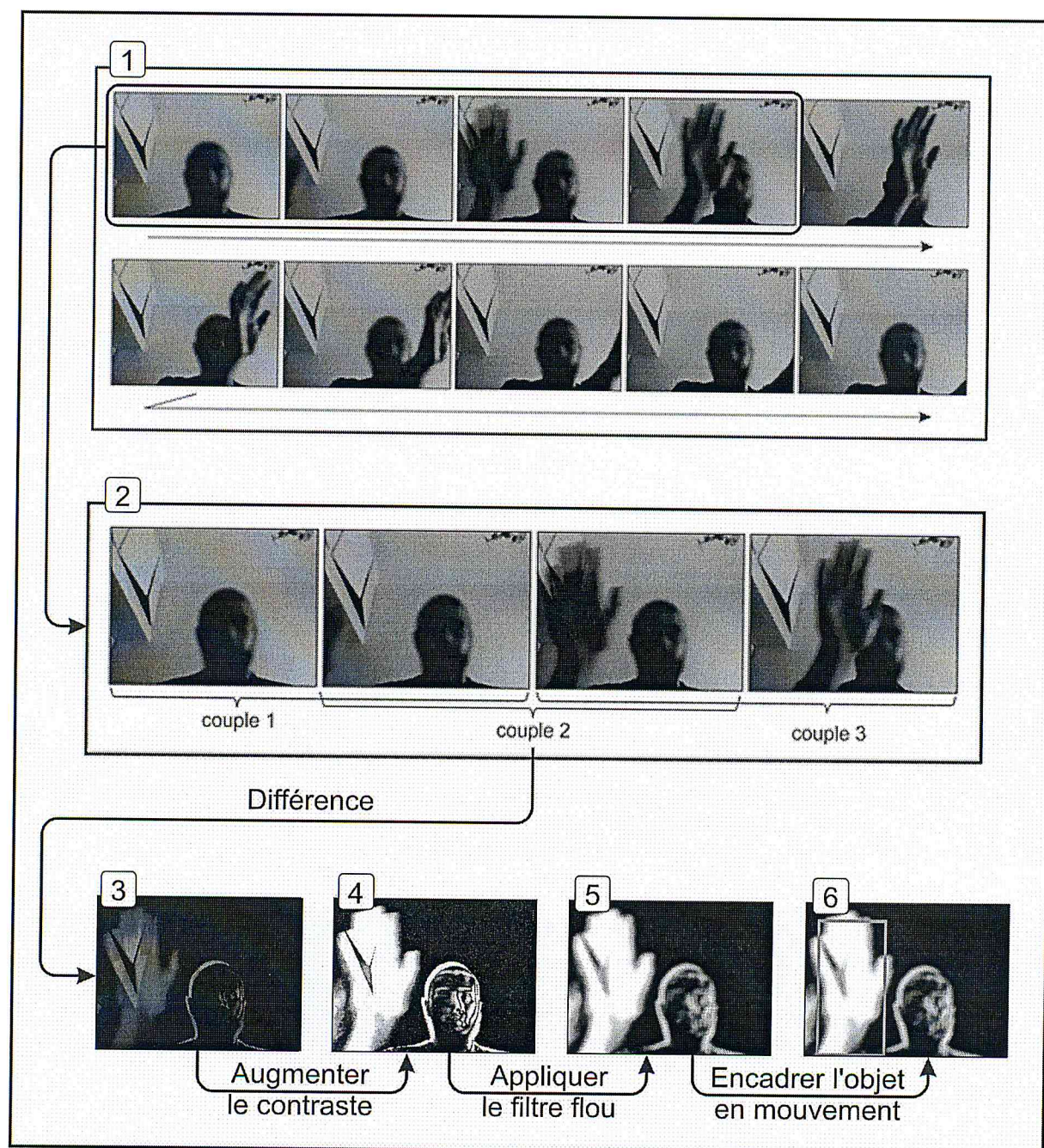


Figure 3.15 : Un exemple d'une reconnaissance d'un geste complet. (Partie 1).

Dans cet exemple illustré dans la figure 3.15, l'étape (1) est une acquisition des images à partir d'une caméra, dont les 4 premières trames sont sélectionnées et couplées dans l'étape (2), puis le couple 2 est pris à part comme exemple en lui appliquant une différence. Le résultat est affiché dans (3), s'en suit une augmentation de contraste, puis une application d'un filtre flou, afin d'encadrer l'objet en mouvement.

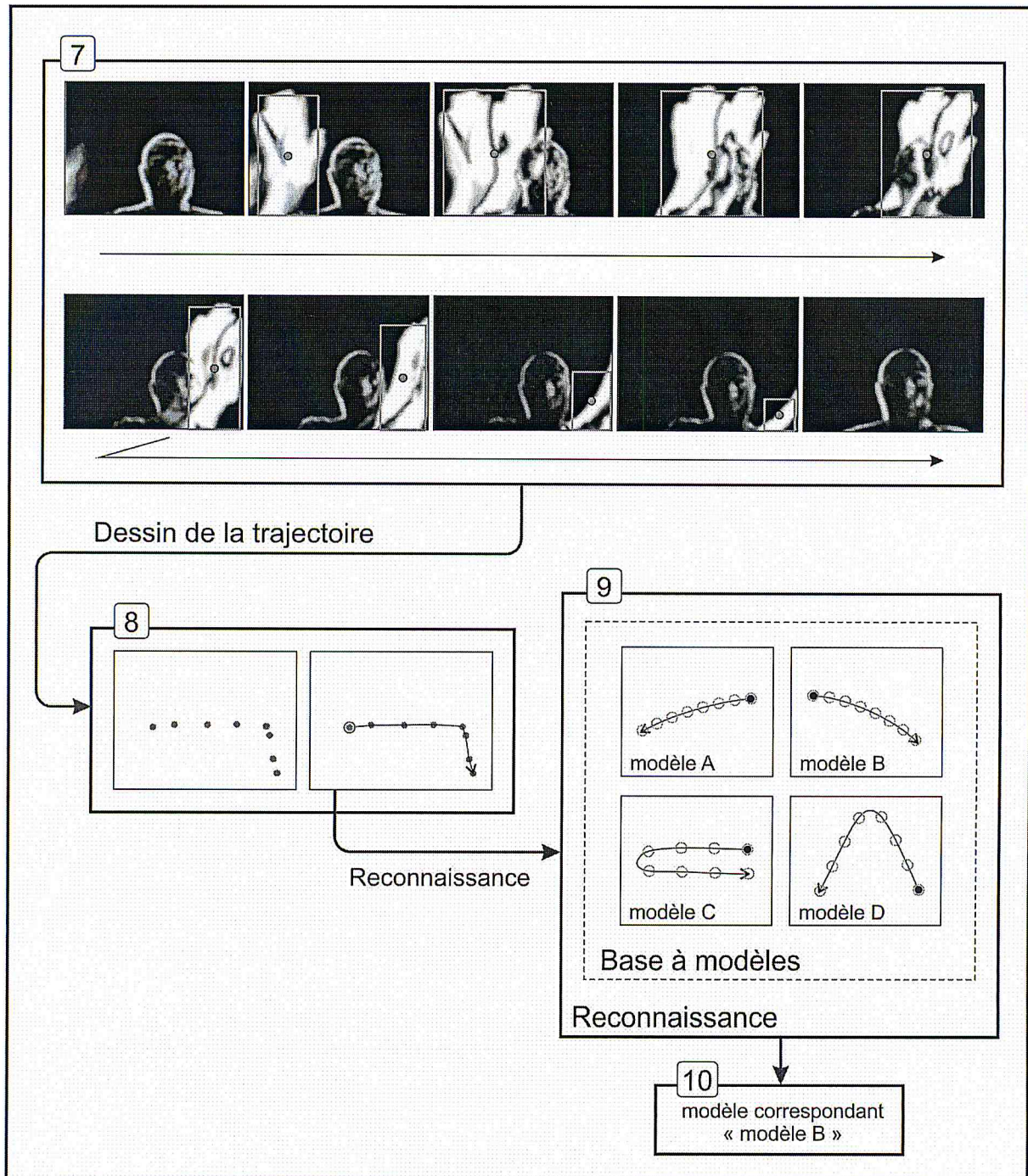


Figure 3.16 : Un exemple d'une reconnaissance d'un geste complet. (Partie 2).

La deuxième partie de l'exemple est illustré dans la figure 3.16. Dans (7) nous appliquons le même processus de la figure 3.15 pour tous les images déjà acquises. Dans (8) nous dessinons la trajectoire pour qu'elle soit comparée avec une base à modèles dans l'étape de reconnaissance (9), afin de déduire le modèle correspondant (10).

3.7. Conclusion

Comme vu dans ce chapitre, les trois étapes : détection, suivi à la trace et reconnaissance ont chacune participé pour la reconnaissance finale du geste détecté et son interprétation.

Pour la méthode de détection qui est basée sur le mouvement, on appliquant plusieurs traitements afin d'éliminer les mouvements insignifiants pour suivre la main dans la scène, puis on a expliqué la méthode du suivi et ces contraintes que nous avons pu les résoudre grâce à notre méthode de «Seconde chance ». Après on a finalisé le processus par une reconnaissance, on utilisant l'algorithme \$1 adapté par notre amélioration, qui compare les trajectoires de la main parcouru avec la base à modèle.

Nous avons dû étudier de nombreuses possibilités afin de mettre au point les approches que nous avons utilisées et nos tests ont été concluants et ceci a abouti finalement dans les applications dont on voit les captures d'écran.

Nous évoquerons dans le prochain chapitre les deux applications développées afin de rendre pragmatique notre étude.

Chapitre 4 : Implémentation

4.1. Introduction

Dans ce chapitre, nous allons voir les différentes applications développées pour implémenter et valider les approches adoptées pour la reconnaissance gestuelle.

Aussi, nous passerons en revue les différents outils, IDE et langages qui ont été utilisés durant nos travaux.

4.2. Application de test pour le débogage

Pour commencer, nous avons développé une application mobile qui se base sur notre approche pour la reconnaissance gestuelle à base de caméra. Cette application nous a aidés à tester, améliorer et valider notre travail à travers des composants divers qui représentent les différentes étapes de cette approche.

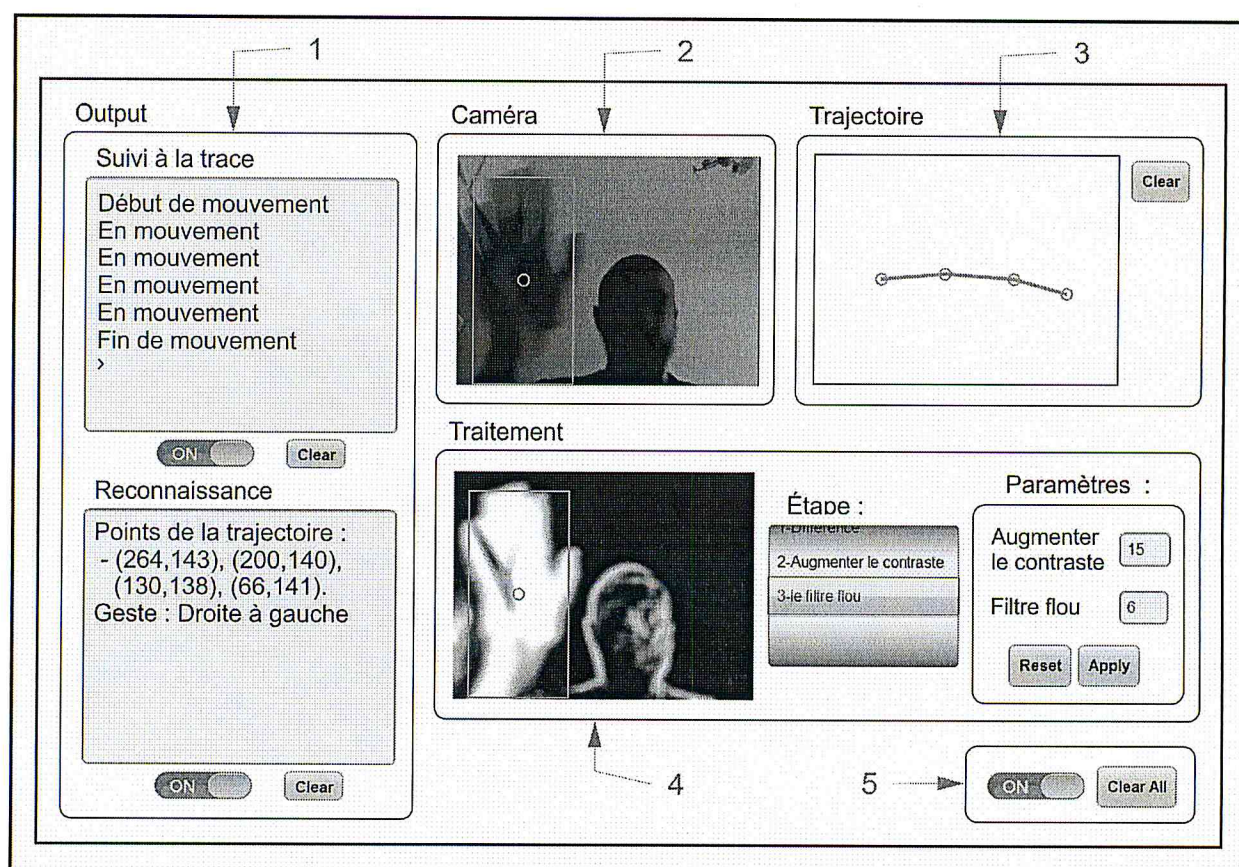


Figure 4.1 : Capture d'écran de notre application sur tablette mobile pour la reconnaissance gestuelle à base de caméra

La figure 4.1 est une capture d'écran de notre application sur tablette, qui se compose de plusieurs zones :

Zone 1 : représente la vidéo acquise par la camera frontale de l'appareil avec l'encadrement de la main en mouvement.

Zone 2 : représente les trajectoires parcourues par la main de l'utilisateur et un bouton « clear » pour les effacer.

Zone 3 : représente les traitements appliqués sur les images acquises par la caméra ainsi que les différentes étapes de détection avec leurs paramètres modifiables.

Zone 4 : représente les deux sorties de l'application : suivi et reconnaissance. Pour le suivi à la trace, on affiche les détails du suivi de l'objet tout au long de son mouvement. Pour la reconnaissance, on affiche sous forme de coordonnées les points de la trajectoire ainsi que son geste associé.

Zone 5 : Un bouton « clear » pour réinitialiser l'application et un bouton commutateur pour activer l'auto-réinitialisation de l'application à chaque début de mouvement.

4.3. Application concrète

Une deuxième application est une sorte d'album photos dans laquelle la navigation se fait par les gestes de la main, cette application est un des exemples à travers lesquels on peut appliquer la reconnaissance gestuelle.

Comme la figure 4.2 l'indique, les gestes que peut exécuter l'utilisateur en face de l'écran de son appareil pour naviguer dans l'application sont :

- Le geste de gauche à droite lui permet d'accéder à l'album suivant ou à l'image suivante, selon le menu.

- Le geste de droite à gauche lui permet d'accéder à l'album suivant ou à l'image suivante, selon le menu.

- Le geste de bas au haut lui permet d'accéder à l'album courant dans le menu des albums.

- En agitant la main devant l'écran, ça permet de revenir au menu précédent.

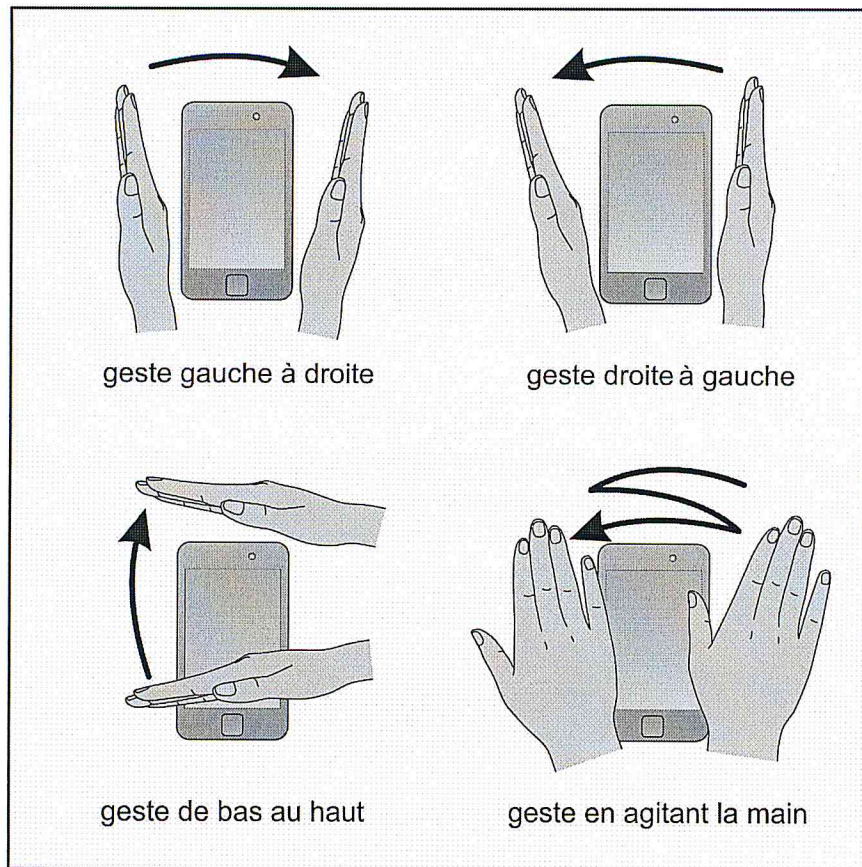


Figure 4.2 : Différents gestes pour naviguer dans l'application

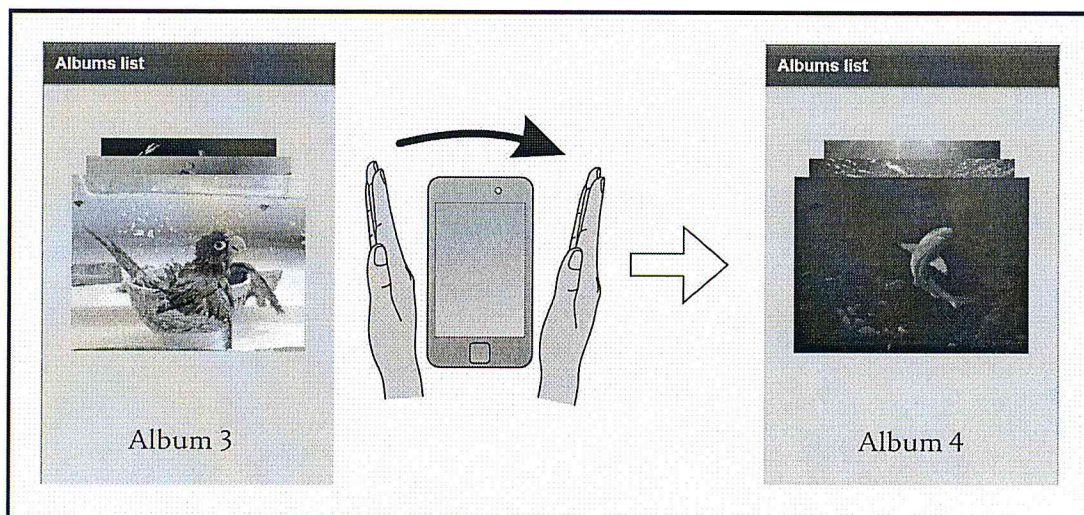


Figure 4.3 : Un geste de gauche à droite pour aller à l'album suivant

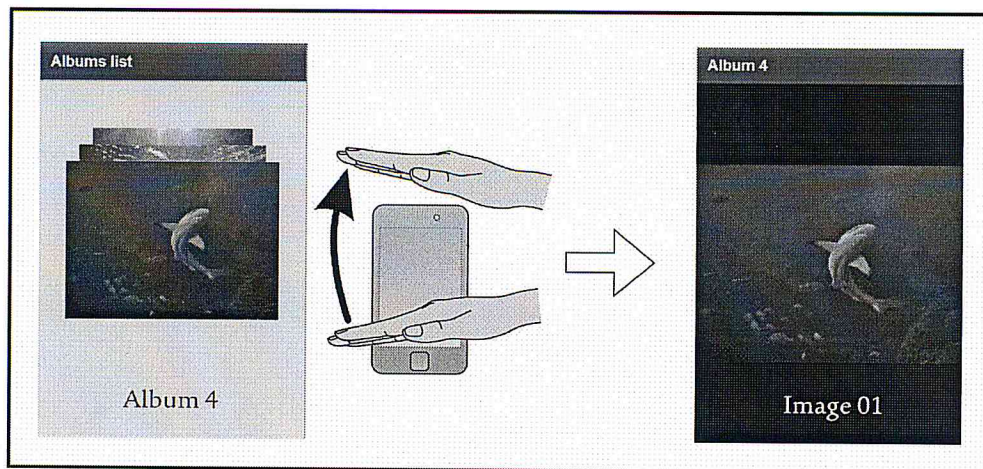


Figure 4.4 : Un geste de bas au haut pour accéder à l'album

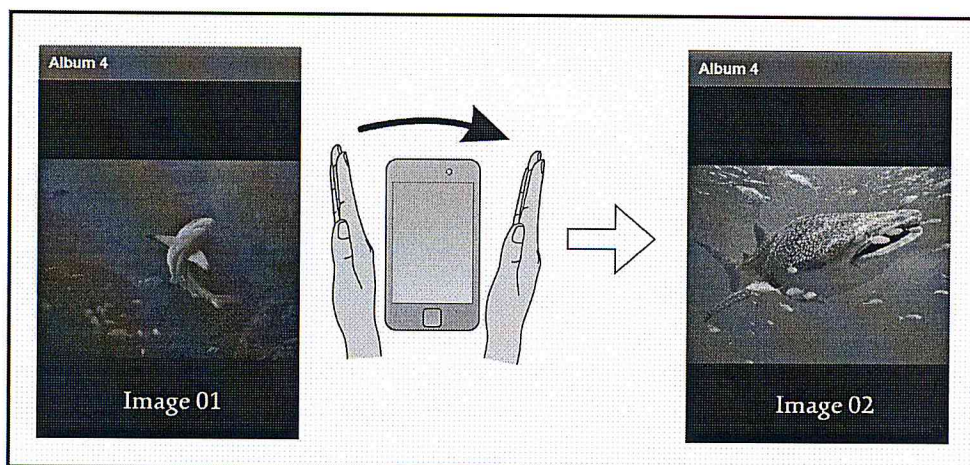


Figure 4.5 : Un geste de gauche à droite pour aller à l'image suivante de l'album

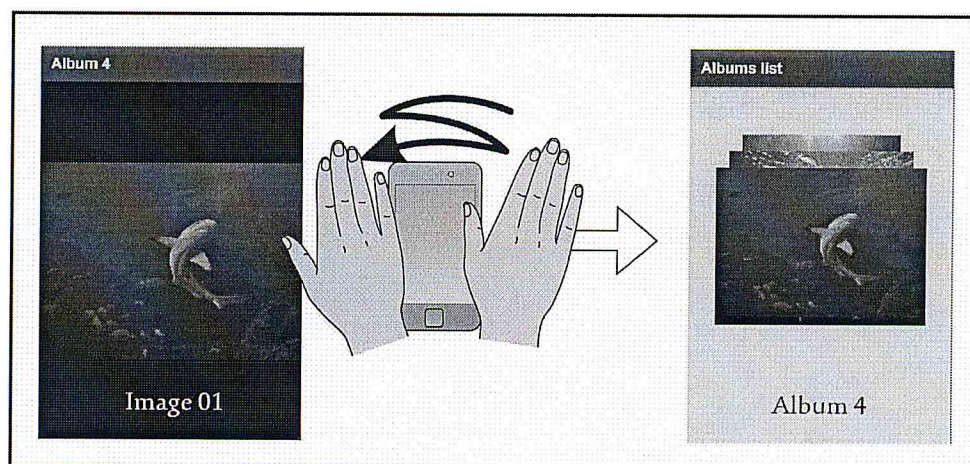


Figure 4.6 : Un geste en agitant la main pour sortir de l'album

4.4. Les outils de travail

Pour valider notre travail, les algorithmes sur lesquels se base notre approche ont été traduits en module *Flex* basé sur le langage de programmation *ActionScript*. En revanche, nos algorithmes peuvent, une fois compris, être implémentés dans n'importe quel langage.

4.4.1. Flex

Apache Flex est un Framework d'application open source, qui permet de créer des applications mobiles pour iOS, Android™ et BlackBerry®, ainsi que des applications de bureau en utilisant le même modèle de programmation.

Le fait qu'un seul code soit compilé sur toutes ces plateformes optimise notre travail. Aussi, son simulateur peut simuler différents modèles de terminaux.

Apache Flex est composé d'un petit nombre de composants différents. Un composant est un compilateur qui combine des documents MXML (layout) avec des fichiers ActionScript. Après la compilation, nous aurons une application de type SWF [69].

4.4.2. Actionscript

ActionScript est un langage de script, orienté objet et prototype, basé sur ECMAScript (une version standardisée de JavaScript), ActionScript et JavaScript partagent une syntaxe semblable.

Les modules basés sur l'*ActionScript* sont des classes, qui peuvent être intégrés et importés dans les applications Flex.

4.4.3. SVN

Nous avons utilisé Apache SVN comme logiciel de gestion de versions pour le développement de nos applications afin d'historiciser le code source relatif aux différentes versions et pouvoir travailler en équipe.

SVN nous a aidés à bien organiser notre travail, et d'éviter les conflits de versions.

4.4.4. Trello

Pour la gestion de projet et pour le partage des tâches entre nous, nous avons utilisé l'outil Trello, application web disponible dans le cloud, très pratique, intuitive et facilitant notre collaboration.

4.5. Conclusion

Notre approche peut être traduite en bibliothèque ou un module pour un environnement de travail, afin de donner la possibilité d'être implémentée facilement par des applications mobiles disposant d'une caméra frontale.

L'exemple de l'album photos est un des exemples à travers lesquels on peut appliquer la reconnaissance gestuelle. D'autres exemples comme un navigateur web, un livre électronique ou même des jeux interactifs, peuvent être de bons exemples pour appliquer cette approche afin d'améliorer l'expérience utilisateur.

CONCLUSION GENERALE

Dans ce travail de recherche, nous avons présenté une nouvelle approche d'interaction avec le mobile, celle qui s'appuie sur la reconnaissance gestuelle en utilisant sa caméra, nous avons découpé notre système en trois étapes.

Pour l'étape de détection, nous avons étudié plusieurs méthodes, ceux qui s'appuient sur la couleur de peau, que nous avons testé sur différents dispositifs mobile, et qu'elle a donné des résultats catastrophiques, en raison de la confusion de la couleur de la main avec celle du visage. La méthode de détection qui se base sur la forme nous a donné aussi des résultats déplorables due au nombre de calcul, ce qui a fait boguée le mobile. Finalement la méthode du mouvement a été choisie comme une méthode de détection, vu sa rapidité, et sa simplicité dans le contexte mobile, d'après nos connaissances nous avons amélioré cette méthode de mouvement (utiliser avant sous la contrainte d'une caméra fixe, ce qui n'est pas dans notre cas), en éliminant les mouvements insignifiants, et ne garder que la main de l'utilisateur afin de le considérer comme l'objet à suivre dans l'étape suivante.

Dans l'étape de suivi ou « tracking », nous avons surmonté cette étape avec notre propre méthode simple et efficace, puisque en générale, les algorithmes et méthodes existantes se base sur la couleur de peau, ou la forme de la main.

Après le suivi de la main de l'utilisateur, on conclut la trajectoire parcourue pour définir le geste correspondant, afin de fournir l'action associée. Pour cette étape nous avons utilisé l'algorithme « *\$I recognizer* » en l'améliorant par la suppression de la phase de

malgré la simplicité, notre système donne des résultats relativement précis et corrects, avec un temps de calcul satisfaisant.

Comme notre approche se base sur la détection de mouvement, le changement de luminance ne se considère pas comme un handicap vu que nous nous basant sur le concept de la différence d'images successives, et c'est très avantageux pour un monde mobile, ou le changement de luminance est omniprésent.

Comme tout système, le nôtre a ces limites, à commencer par l'objet détectable, dans notre système, comme il est basé sur la méthode de détection par mouvement, chaque mouvement brusque est considéré comme la main de l'utilisateur, par exemple : si on bouge la tête brusquement de gauche à droite, le système va le considéré comme un geste de la main allant de gauche à droite.

Notre passion pour le mobile nous a conduit à réfléchir et à travailler sur le domaine de la reconnaissance gestuelle en utilisant la caméra afin d'offrir une nouvelle expérience utilisateur plus naturelle et plus intuitive. Nous envisageons pour le futur de rendre nos travaux accessibles à d'autres utilisateurs sous forme de *framework* afin qu'ils puissent implémenter cette technologie de reconnaissance plus facilement et ne pas réinventer la roue. Depuis l'arrivée des appareils mobiles dits " téléphones intelligents", le nombre d'applications et de développeurs a explosé, et la reconnaissance gestuelle a un avenir indéniable.

Un des axes d'amélioration sur lequel on souhaiterait travailler est l'apprentissage automatique, domaine très proche de l'intelligence artificielle afin d'automatiser le processus d'enrichissement de la base de modèles très rapidement.

Bibliographie

- [1] «<http://www.it-business.fr/tablettes-tactiles-entreprises-tendance/>».
- [2] «Evolution Of Mobile Phones: 1995 – 2012 By Rahul Chowdhury : <http://www.hongkiat.com/blog/evolution-of-mobile-phones/>».
- [3] Ying Wu, Thomas S. Huang, and N. Mathews. Vision-based gesture recognition : A review. In Lecture Notes in Computer Science, pages 103,115. Springer..
- [4] R.E. Axtell. Gestures: The do's and taboos of body language around the world. Wiley, 1991..
- [5] LIVING IN A SYMBOLIC WORLD : <http://www.trinity.edu/mkearl/socpsy-4.html>.
- [6] M. Krueger. Environmental technology: Making the real world virtual. Communications of the ACM, 36:36{37, 1993..
- [7] <http://blog.seattlepi.com/digitaljoystick/2009/06/01/e3-2009-microsoft-at-e3-several-metric-tons-of-press-releaseapalloza/>, (18/03/2013).
- [8] <http://www.anandtech.com/show/4057/microsoft-kinect-the-anandtech-review/3>, (18/03/2013).
- [9] <http://www.t3.com/features/exclusive-how-does-microsoft-xbox-kinect-work>, (18/03/2013).
- [10] Kinect documentation.
- [11] R. Herpers, G. Verghese, K. Darcourt. An active stereo vision system for recognition of faces and related hand gestures. In Int. Conf. on Audio- and Video-based Biometric Person Authentication, pages 217,223, Washington, D. C., 1999..
- [12] T. Kurata, T. Okuma, M. Kouroggi, and K. Sakaue. The hand mouse: Gmm hand-color classification and mean shift tracking. In Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, pages 119{124, Vancouver, BC.
- [13] D. Saxe and R. Foulds. Toward robust skin identification in video images. In IEEE Int.

- Conf. on Automatic Face and Gesture Recognition, pages 379,384, 1996, 1996..
- [14] D. Chai and K. Ngan. Locating the facial region of a head and-shoulders color image. In IEEE Int. Conference on Automatic Face and Gesture Recognition, pages 124{129, Piscataway, NJ, 1998..
- [15] A. A. Argyros and M. I. A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In Proc. European Conference on Computer Vision, pages 368,379, Prague, Czech Republic, May 2004..
- [16] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. International Journal of Computer Vision, 46(1):81,96, 2002..
- [17] X. Zhu, J. Yang, and A. Waibel. Segmenting hands of arbitrary color. In Proc. International Conference on Automatic Face and Gesture Recognition (FG), pages 446,455, Grenoble, France, March 2000..
- [18] D. Gavrila and L. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In Proc. IEEE Computer Vision and Pattern Recognition (CVPR), pages 73,80, 1996, 1996..
- [19] S. M. Yoon and H. Kim. Real-time multiple people detection using skin color, motion and appearance information. In Proc. IEEE International Workshop on Robot and Human In teractive Communication (ROMAN), pages 331 {334, Kurashiki Okayama Japan, Sep.
- [20] J. P. Zhou and J. Hoang. Real time robust human detection and tracking system. In Proc. IEEE Computer Vision and Pattern Recognition (CVPR), pages III: 149{149, 2005..
- [21] A. Utsumi and J. Ohya. Direct manipulation interface using multiple cameras for hand gesture recognition. In SIGGRAPH, page 112, 1997. 4 Summary 54.
- [22] J. Kuch and T. Huang. Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. In Proc. International Conference on Computer Vision (ICCV), pages 666{671, 1995..
- [23] A. Downton and H. Drouet. Image analysis for model-based sign language coding. In

- Int. Conf. Image Analysis and Processing, pages 637{644, 1991..
- [24] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509,522, 2002..
- [25] A. A. Argyros and M. I. A. Lourakis. Vision-based interpretation of hand gestures for remote control of a computer mouse. In *ECCV Workshop on HCI*, pages 40{51, Graz, Austria, May 2006..
- [26] R. Vaillant and D. Darmon. Vision-based hand pose estimation. In *Int. Workshop on Automatic Face and Gesture Recognition*, pages 356{361, Zurich, Switzerland, 1995..
- [27] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. International Conference on Computer Vision (ICCV)*, pages 612{617, 1995..
- [28] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *Proc. International Conference on Computer Vision (ICCV)*, pages 426{432, Vancouver, Canada, July 2001..
- [29] B. Stenger, R. Mendonca, and R. Cippola. Model-based 3D tracking of an articulated hand. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 126,133, Hawaii, 2002..
- [30] L. Song and M. Takatsuka. Real-time 3D finger pointing for an augmented desk. In *Australasian conference on User interface*, volume 40, pages 99{108, Newcastle, Australia, 2005..
- [31] C. Jennings. Robust finger tracking with multiple cameras. In *IEEE workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 152{160, Corfu, Greece, 1999. 4 Summary 46.
- [32] A. A. Argyros and M. I. A. Lourakis. Binocular hand tracking and reconstruction based on 2D shape matching. In *Proc. International Conference on Pattern Recognition (ICPR)*, Hong-Kong, China, 2006..
- [33] M. Yin, X.and Xie. Estimation of the fundamental matrix from uncelebrated stereo hand

- images for 3D hand gesture recognition. *Pattern Recognition*, 36(3):567{584, 2003..
- [34] A. Meyering and H. Ritter. Learning to recognize 3D-hand postures from perspective pixel images. In *Artificial Neural Networks II*, pages 821 {824. Elsevier Science Publishers, 1992..
- [35] D.S. Zhang et G. Lu, Segmentation of Moving Objects in Image Sequence: A Review, *Circuits, Systems and Signal Processing (Special Issue on Multimedia Communication Services)*, volume 20(2), p 143-183, 2001..
- [36] S. Andra, O. Al-Kofahi, R.J. Radke, et B. Roysam, Image Change Detection Algorithm: A systematic.
- [37] M. Piccardi, Background subtraction techniques: a review, *IEEE SMC 2004 International Conference on Systems, Man and Cybernetics*, Octobre 2004..
- [38] M.J. Black et P. Anandan, The robust estimation of multiple motions : Parametric and piecewise-smooth flow fields, *CVIU*, volume 63, p. 75-104, Janvier 1996..
- [39] S. Jehan-Besson et M. Barlaud, DREAM2S: Deformable Regions Driven by an Eulerian Accurate Minimization Method for Image and Video Segmentation, *IJCV*, volume 53(1), p. 45-70, 2003..
- [40] T.H. Kim, Y.S. Moon, A New Flat Zone Filtering Using Morphological Reconstruction Based on the Size and Contrast, *VLBV*, 1999.
- [41] W. Freeman and C. Weissman. Television control by hand gestures. In *Int. Workshop on Automatic Face and Gesture Recognition*, pages 179{183, Zurich, Switzerland, 1995..
- [42] F. Quek. Eyes in the interface. *Image and Vision Computing*, 13(6):511{525, 1995..
- [43] Q. Yuan, S. Sclaro®, and V. Athitsos. Automatic 2D hand tracking in video sequences. In *IEEE Workshop on Applications of Computer Vision*, pages 250{256, 1995..
- [44] J. Crowley, F. Berard, and J. Coutaz. Finger tracking as an input device for augmented reality. In *International Workshop on Gesture and Face Recognition*, Zurich, June 1995..
- [45] R. O'Hagan and A. Zelinsky. Finger Track - a robust and realtime gesture interface. In *Australian Joint Conference on Artificial Intelligence*, pages 475,484, Perth, Australia,

November 1997..

- [46] T. Darrell, I. Essa, and A. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(12):1236, 1242, 1996..
- [47] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proc. IEEE Com-*.
- [48] T. F. Cootes and Taylor C. J. Active shape models – smart snakes. In *British Machine Vision Conference*, pages 266,275, 1992..
- [49] T. F. Cootes, Taylor C. J., Cooper D. H., and J. Graham. Active shape models - their training and applications. *Computer Vision and Image Understanding*, 61(1):38,59, 1995..
- [50] E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35-42, 1960..
- [51] Nguyen Dang Binh, Enokida Shuichi, Toshiaki Ejima , Real-Time Hand Tracking and Gesture Recognition System, Kyushu Institute of Technology 680-4, Kawazu, Iizuka, Fukuoka 820, JAPAN, December 2005..
- [52] M. Breig and M. Kohler. Motion detection and tracking under constraint of pan-tilt cameras for vision-based human computer interaction. Technical Report 689, Informatik VII, University of Dortmund/Germany, August 1998..
- [53] A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 473,478, Colorado, 1999..
- [54] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. PFinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780,785, 1997..
- [55] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 568{574, Puerto Rico,

1997..

- [56] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 5,28, 1998..
- [57] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *IEEE Int. Conf. Automatic Face and Gesture Recognition*, pages 140{145, Killington, VT, 1996..
- [58] J. Lin, Y. Wu, and T. S. Huang. Capturing human hand motion in image sequences. In *Proc. IEEE workshop on Motion and Video Computing*, pages 99,104, 2002..
- [59] H. Birk, T. B. Moeslund, and C. B. Madsen. Real-time recognition of hand alphabet gestures using principal component analysis. In *Proc. Scandinavian Conference on Image Analysis*, Lappeenranta, Finland, June 1997..
- [60] D. J. Mapes and M. J. Moshell. A two-handed interface for object manipulation in virtual environments. *PRESENSE: Teleoperators and Virtual Environments*, 4(4):403{416, 1995..
- [61] Yiqiang CHEN , Wen GAO , Jiyong MA, Hand Gesture Recognition Based on Decision Tree, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, Octobre 2010.
- [62] J. Mackie and B. McCane, Finger Detection with Decision Trees, University of Otago, Dept. of Computer Science.
- [63] Klimis Symeonidis, Hand Gesture Recognition Using Neural Networks, School of Electronic and Electrical Engineering, August 23, 2000.
- [64] miss. Shweta K. Yewale, mr. pankaj k. bharne, artificial neural network approach for hand gesture recognition, institute of technology & research, badnera, sipna college of engg. & technology, amravati..
- [65] Rafiqul Z. Khan, Noor A. Ibraheem, Vision Based Gesture Recognition Using Neural Networks Approaches: A Review, Faculty of Science /Department of Computer Science/ A.M.U. Aligarh, india, 2002..

- [66] François Denis et Rémi Gilleron, APPRENTISSAGE À PARTIR D'EXEMPLES, Notes de cours..
- [67] Wobbrock, J.O., Wilson, A.D. and Li, Y. (2007). Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)..
- [68] Jacob O. Wobbrock, \$1 Unistroke Recognizer, <http://depts.washington.edu/aimgroup/proj/dollar/>, (06/03/2013).
- [69] About Apache Flex, <http://flex.apache.org/about-what.html>, 12/04/2013.
- [70] M. Cote, P. Payeur, and G. Comeau. Comparative study of adaptive segmentation techniques for gesture analysis in unconstrained environments. In IEEE Int. Workshop on Imaging Systems and Techniques, pages 28,33, 2006..
- [71] L. Sigal, S. Sclaro®, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. IEEE Trans. Pattern Analysis and Machine Intelligence, 26(7):862,877, 2004..
- [72] T. Darrell and A. Pentland. Space-time gestures. In Proc. IEEE Computer Vision and Pattern Recognition (CVPR), pages 335,340, New York, NY, 1993..
- [73] H. Fillbrandt, S. Akyol, and K. F. Kraiss. Extraction of 3D hand shape and posture from images sequences from sign language recognition. In Proc. International Workshop on Analysis and Modeling of Faces and Gestures, pages 181,186, Nice, France, O.
- [74] A. Wilson and A. Bobick. Learning visual behavior for gesture analysis. In IEEE Symposium on Computer Vision, Coral Gables, FL, 1995..
- [75] G. Rigoll, A. Kosmala, and M. Schusterm. A new approach to video sequence recognition based on statistical methods. In Proc. International Conference on Image Processing (ICIP),.
- [76] [ML97] P. Morguet and M. K. Lang. A universal HMM-based approach to image sequence classification. In Proc. International Conference on Image Processing (ICIP), pages 146,149, 1997..

- [77] G. Rigoll, A. Kosmala, and S. Eickeler. High performance realtime gesture recognition using hidden Markov models. Lecture Notes in Computer Science, 1998..
- [78] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates. In IEEE Workshop on Applications of Computer Vision, pages 39{42, Sarasota, FL, 1996..
- [80] John MacCormick, How does the Kinect work ?, 2012.
- [81] <http://allaboutgalaxys4.com/galaxy-s4-features-explained/air-gesture/>, (13/06/2013).
- [83] Deep J. Shah, Motion Based Bird Sensing Using Frame Differencing.
- [84] Galaxy S4 Features: Air Gesture <http://allaboutgalaxys4.com/galaxy-s4-features-explained/air-gesture/>, (13/06/2013)..

Glossaire

Une image : un ensemble de points ou pixels, chaque pixel contient trois informations essentielles, la valeur en octets de la couleur rouge, vert et bleue.

Filtrage d'image : Le principal objectif du filtrage est de prétraiter une image afin que l'image filtrée soit plus adaptée que l'image originale pour une application spécifique.

Filtre de flou (Blur) : Le flou est une opération très puissante utilisée dans le traitement d'images. Le filtre flou concerne le calcul des moyennes pondérées des zones de pixels d'une image pour chaque pixel. Une autre façon de voir les choses est que chaque pixel de l'image de destination est constitué d'un mélange de pixels voisins de l'image source.

La différence de trames : La différence de trames (ou différences d'images successives) est une technique où l'ordinateur vérifie la différence entre deux trames. Si les pixels changent, ce qu'il y a bien un mouvement. Nous appliquons des effets de flou et de seuil pour réduire les mouvements insignifiants.

Zone de délimitation : ou un cadre de délimitation est une zone rectangulaire couvrant l'ensemble des arêtes de l'objet détecté à l'intérieur. La taille de cette zone est déterminée par la taille de l'objet à recouvrir.

Centre de gravité : Un centre de gravité est défini comme étant le centre d'un objet détecté en mouvement.

Filtre de Kalman : Le filtre de Kalman est un filtre qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées.

Espace colorimétrique : Un espace colorimétrique est une représentation mathématique d'un ensemble de couleurs, Tous les espaces peuvent être dérivés de l'information RGB.

Chromaticité : est la combinaison de la teinte qui est la forme pure d'une couleur, avec la saturation qui décrit l'intensité d'une couleur, c'est une propriété à deux dimensions.

Luminance : est la propriété qui décrit la "luminosité" de la lumière.

Occlusion : une occlusion désignant une ombre de contact entre 2 objets.

Histogramme : est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse.

Apprentissage automatique (machine-learning en anglais) : fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage.

Traitement d'images : On désigne l'ensemble des techniques permettant de modifier une image numérique dans le but de l'améliorer ou d'en extraire des informations.

Classification : ou système de classification est un système organisé et hiérarchisé de catégorisation d'objets

Framework : est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un squelette de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle.

Logiciels de gestion de versions : (ou VCS en anglais, pour Version Control System) est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus.

