

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

BOUMARAF YANISS ZEMOURI YUCEF

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel



Thème
***Systeme de reconnaissance de gestes humain :
Application à la vidéo protection***

Mr. A.CHERIF ZAHAR
Mr. A.CHEMCHAM
Mr. KAMECHE ABDALAH
Mr. DJEKOUN OUALID

Président
Examineur
Promoteur
Encadreur



Promotion
2015 / 2016

Dédicaces

A :

Nos chers parents

Nos frères et sœurs

Toute Notre famille

Tous nos amis

On dédie ce mémoire

BOUMARAF YANISS

ZEMOURI YUCEF

Remerciement

On remercie en premier lieu le tout puissant de nous avoir donné la chance, le courage et les moyens pour l'accomplissement de ce projet.

On remercie cordialement, Nos encadreurs Mr Djekoune A. Oualid, Mme Amara Kahina de nous avoir proposé ce thème, de nous avoir guidé et suivi durant l'accomplissement de ce travail.

On remercie tout particulièrement Mr Kameche AbdAlah, enseignant à l'université USDB, pour nous avoir aidés et épaulé durant toute cette année, de nous avoir guidé et soutenu dans la réalisation du projet, merci.

On remercie tous les membres du jury d'avoir accepté d'examiner notre travail.

Au terme de ce mémoire, On tient à remercier toute personne ayant contribué à l'accomplissement de ce travail de près ou de loin et qui nous ont été bénéfique durant notre parcours pour la réalisation de ce travail.

Finalement, On remercie l'ensemble des enseignants du département d'informatique pour les efforts qu'ils fournissent pour notre réussite dans nos études.

BOUMARAF YANISS

ZEMOURI YOUCEF

LISTE DES FIGURES

Fig.I.1: Caméras de surveillance	17
Fig.I.2: Schéma de vidéosurveillance analogique avec magnétoscope traditionnels	19
Fig.I.3: Schéma de vidéosurveillance analogique avec enregistreur numérique	19
Fig.I.4: Schéma de vidéosurveillance analogique avec enregistreur numérique réseau	20
Fig.I.5: Schéma de vidéosurveillance sur IP avec serveur vidéo.....	20
Fig.I.6: Schéma de vidéosurveillance sur IP avec caméras réseau.....	21
Fig.I.7: Caméra IP [6]. (a) : face avant (b) : face arrière	22
Fig.I.8: Schéma illustrant l'utilisation des encodeurs et décodeurs vidéo dans un système	24
Fig.I.9: Schéma illustrant les différentes parties d'un système de vidéosurveillance qui peuvent être équipées des fonctionnalités d'intelligence	27
Fig.I.10: Système centralisé à base de DVR	29
Fig.I.11: Système centralisé à base d'un PC serveur.....	29
Fig.I.12: Système distribué, les fonctionnalités d'intelligences sont aux extrémités de réseau (à gauche cas de caméras IP, à droite cas de caméras analogiques)	30
Fig.II.1: Un Système de Détection et de Reconnaissance du Geste.....	39
Fig.II.2: Schéma d'un système de reconnaissance de geste.....	39
Fig.II.3: Capteurs grand public. (a) Caméra de profondeur Microsoft Kinect, (b) Bracelet à centrale inertielle MoveaMotionPod, (c) Manette à capteur magnétique Razer Hydra et modification possible	41
Fig.II.4: L'apprentissage supervisé	43
Fig.II.5: Techniques possibles de reconnaissance de gestes	43
Fig.II.6: Association de classe de gestes a un geste continu, extrait de (a)apprentissage de trois gestes de référence. (b) reproduction d'un geste en continu en entrée du modèle. (c) extraction des classes de gestes par le système de reconnaissance	44
Fig.III.1: Schéma d'un système de reconnaissance de geste	51
Fig.III.2: Exemple de document XML	52
Fig.III.3: Exemples de gestes pris en compte par notre base de données	53
Fig.III.4: La face avant de la Kinect	55

Fig.III.5: Les composants de la Kinect	57
Fig.III.6: Positions des articulations du corps humain détectés	60
Fig.III.7: L'exemple du geste « Glisser main droite vers la gauche »	61
Fig.III.8: La classe des différentes relations	61
Fig.III.9: Schéma du programme qui gère les gestes	62
Fig.III.10: Exemples sur des postures	63
Fig.III.11: Schéma du programme qui gère les postures et les scénarios	64
Fig.III.12: Exemples pour les scénarios	65
Fig.III.13: Schéma du programme qui gère les postures et les scénarios	66
Fig.III.14: La matrice des séquences	69
Fig.III.15: La méthode de calcul de la matrice de Needleman	69
Fig.III.16: Bords de la matrice	70
Fig.III.17: Explication des calculs de la matrice de Needleman.....	71
Fig.IV.1: Schéma global du système	74
Fig.IV.2: Microsoft Visual Studio	76
Fig.IV.3: Fenêtre principale	77
Fig.IV.4: Exemple qui montre l'affichage RGB et le squelette	78
Fig.IV.5: Nouveau geste dynamique	79
Fig.IV.6: Conditions du geste.....	79
Fig.IV.7: Nouvelle posture	80
Fig.IV.8: Conditions des postures	81
Fig.IV.9: Création de scénario.....	81
Fig.IV.10: Posture existante pour la création du scénario	82
Fig.IV.11: Scénarios utilisés pour tester les deux mesures de similarité	83
Fig.IV.12: Posture utilisée pour tester le cas du distributeur de billet	87
Fig.IV.13: Scénarios utilisés pour tester le cas du distributeur de billet.....	87
Fig.IV.14 : Individu qui est entrain de mettre la carte dans le distributeur de billet.....	88
Fig.IV.15 : Individu qui est entrain de saisir ces informations	88
Fig.IV.16 : Individu qui est entrain de sélectionner le montant d'argent.....	89
Fig.IV.17 : Individu qui est entrain de retirer de l'argent.....	89
Fig.IV.18 : Cas où aucun mouvement n'a été effectué	90

LISTE DES TABLEAUX

Tab.III.1 : Tableau de répétions des caractères dans les deux séquences	68
Tab.IV.1 :Tableau comparatif des résultats obtenus des deux méthodes de similarité	85
Tab.IV.2 : Tableau interprétant la validité des résultats obtenus par rapport au seuil.....	86

TABLE DES MATIERES

Introduction Générale	12
------------------------------------	----

Chapitre I : Vidéosurveillance et vidéosurveillance intelligentes

1. Introduction	16
2. Les systèmes de vidéosurveillance	17
2.1- Définition	17
2.2- Apparition.....	17
2.3- Evolution des systèmes de vidéosurveillance	18
3. Eléments constituant un système de vidéosurveillance	21
3.1-Caméra	21
3.2-Caméra IP	22
3.3- Enregistreur vidéo	22
3.4-Encodeur vidéo	23
3.5-Décodeur vidéo	23
4. Composantes fonctionnel d'un système de vidéosurveillance	24
5. Les systèmes de vidéosurveillance intelligente	26
5.1-Définition	26
5.2-L'importance de la vidéosurveillance intelligente.....	27
5.3-Avantage de la vidéosurveillance intelligente	27
5.4-Architecture de la vidéosurveillance intelligente.....	28
5.4.1-Architecture centralisée	28
5.4.2-Architecture distribuée.....	30
6. Domaines d'application des systèmes de vidéosurveillance	30
7. Conclusion	32

Chapitre II : La reconnaissance de geste

1. Introduction	34
2. Définition du geste	35
3. Types de geste	36
4. Styles de gestes	36
5. Fonctionnalités du geste	37
6. La reconnaissance de geste	38
7. Système de reconnaissance de gestes	39
7.1-Gestion d'un système d'apprentissage et de reconnaissance du geste pour l'interaction	40
7.1.1-Acquisition.....	40
7.1.2-Représentation des données : traitement et descripteurs gestuels.....	41
7.1.3-Classification	42

7.1.4-Interprétation.....	45
7.2- Application des systèmes de reconnaissance de gestes	46
8. Conclusion.....	48

Chapitre III : Conception et réalisation

1. Introduction	50
2. Démarche adopté pour concevoir le système de reconnaissance de gestes	51
2.1- Création de notre base de données des gestes.....	51
2.2- Les étapes classiques d'un système de reconnaissance de gestes	54
2.2.1-Etape 1 : Acquisition du geste	54
2.2.2-Etape 2 : Segmentation et extraction des caractéristiques	59
2.2.3-Etape 3 : Classification	67
3. Conclusion	72

Chapitre IV : Implémentation et résultats

1. Introduction	74
2. Environnement de développement	74
2.1- Langage de programmation C#	74
2.2-Microsoft.Net.....	75
2.3- Outils de développement Microsoft Visual Studio.....	75
2.4- WinForms et WPF	76
3. Présentation de l'application.....	77
3.1- Interface fenêtre principale	77
3.2- Interface nouveau geste dynamique.....	78
3.3 - Interface nouvelle posture	80
3.4- Interface création scénario	81
4. Résultats des tests effectués	83
5. Conclusion.....	91

Conclusion Générale	93
----------------------------------	-----------

Limites et Perspectives	96
--------------------------------------	-----------

Bibliographie	98
----------------------------	-----------

Résumé

Ce projet se place dans le domaine de la vidéo protection et plus précisément le domaine de la reconnaissance des gestes humain.

L'objectif de notre projet est la conception et la réalisation d'un système de reconnaissance de geste du corps humain en utilisant le capteur Kinect et le langage de programmation C#.

Notre application suit des étapes qui englobent le processus de reconnaissance de gestes en commençant par l'acquisition des données de l'environnement. La seconde est l'analyse et la détermination de l'état ou la posture des individus détectés selon des règles bien établies, et enfin, la dernière étape est la classification de ses résultats.

Cette application facilitera le travail des agents manipulant des systèmes de la vidéo protection pour CDTA (Centre de Développement Technologies Avancées) ou autres institutions possédant les mêmes systèmes.

Mots clés: Geste, Système de reconnaissance de geste, Vidéo protection, Capteur de vision, Kinect, Mesure de similarité, Similarité cosinus, Needleman-Wunsch.

Abstract

The project is held in the domain of video protection and precisely the domain of human gesture recognition.

The objective of our project is to make a system of human body gesture recognition using the Kinect sensor and C# programming language.

Our application follows steps which include the gesture recognition process starting with the acquisition of environmental data. The second step is the analysis and the determination of the detected individuals' stat or posture according to rules well established, and finally, the last step is the classification of the results.

This application will facilitate the work of agents who use video protection systems for CDTA (Centre de Développement Technologies Avancées, or Advanced Technology Development Center in English) or other institutions with the same systems.

Keywords: Gesture, Gesture recognition system, Video protection, Sensor, Kinect, Measuring similarity, Cosine similarity, Needleman-Wunsch.

ملخص

يدخل هذا المشروع ضمن ميدان فيديو الحماية و بالخصوص ميدان التعرف على حركة الجسم

البشري.

و يهدف هذا المشروع إلى خلق و تحقيق نظام التعرف على حركة الجسم البشري و ذلك باستخدام رؤية كينيكيت.

و يشمل تطبيقنا هذا على ثلاث مراحل للتعرف على حركة الجسم البشري و المتمثلة في: مرحلة اكتساب المعطيات من المحيط و المرحلة الثانية هي تحليل و تحديد حالة أو وضعية الشخص المكشوف عن طريق قواعد قائمة بذاتها و أخيرا المرحلة الثالثة و التي تتمثل في تصنيف هذه النتائج.

سيسهل هذا التطبيق عمل أعوان المستخدمين لنظام فيديو الحماية لمركز تطوير التكنولوجيا المتقدمة أو مختلف المؤسسات التي تتبع نفس النظام.

الكلمات الرئيسية: حركة، نظام التعرف على الحركة، فيديو الحماية، أجهزة رؤية كينيكيت، قياس التشابه جب تمام، تشابه القياس، نيدلمان - ونش

Introduction Générale

Le désir de pouvoir interagir avec un ordinateur de manière intuitive et naturelle est grandissant. Dans cette optique, la recherche en reconnaissance de gestes propose de développer des systèmes capables de modéliser, d'analyser et de reconnaître les gestes d'un utilisateur humain.

Les gestes et la reconnaissance des gestes sont des termes de plus en plus rencontrés dans le domaine des jeux vidéo, avec l'apparition des nouveaux contrôleurs développés par les constructeurs de consoles de jeux. L'intégration du contrôle à travers des gestes dans des logiciels embarqués ou qui contrôlent des objets à distance pourrait constituer un grand pas en avant dans des domaines comme la robotique ou la domotique.

La reconnaissance de gestes humains regroupe l'ensemble des techniques visant à capturer des informations caractérisant les mouvements d'un corps humain dans l'espace et à détecter/reconnaître des gestes significatif d'une action donnée, d'une intention ou caractérisant une expression spécifique. C'est un des nombreux défis posés dans le domaine de la vision par les machines ainsi dans le domaine de la vidéo protection et qui a gagné en attention au cours des dernières décennies avec les évolutions tant technologiques (caméra à capteur de profondeur, par exemple Kinect) que les méthodes et outils développés par les communautés issues du traitement du signal et de l'apprentissage artificiel.

Les systèmes de reconnaissance des gestes peuvent dépendre des dispositifs de capture utilisé (vidéo 2D, vidéo à capteur de profondeur type Kinect, gants de données, etc.); afin de détecter et d'extraire la forme du corps humain de son environnement, il faut traiter d'abord ces informations de façon à extraire ce qui caractérise l'action humaine (extraction de descripteur), détecter/reconnaître l'action elle-même (classification) et ceci pour assurer l'interaction en temps réel.

La nature des données initiales dépendent du dispositif de capture utilisé et des prétraitements déjà effectués sur ces dernières.

Actuellement les recherches s'orientent vers de nouveaux ports d'interaction, qui offrent un moyen de communication plus naturel, direct et efficace en se basant sur les canaux de communication sensorielle humaine.

Nous essayons à travers ce travail de trouver une méthode facile de reconnaissance de gestes pour une application de vidéo-protection, afin de faciliter le travail des agents de sécurité. Nous utilisons le capteur Kinect pour identifier les individus, extraire la forme du corps humain de son

environnement, puis traiter ces informations pour caractériser les différents gestes statiques ou dynamiques en temps réel et ensuite les classer selon leurs catégories.

Le défi principal dans la reconnaissance des gestes repose dans la compréhension des caractères des gestes. L'exploration en utilisant ces caractères dans la reconnaissance de gestes est une phase cruciale pour atteindre les résultats désirés. Le but est de concevoir un système de surveillance qui, au lieu de se contenter d'un enregistrement passif des séquences, nécessitant l'interprétation d'un opérateur humain, il sera capable d'analyser automatiquement et en temps réel le contenu des vidéos pour attirer l'attention sur des événements anormaux (mouvements suspects...).

Dans ce projet nous voulons proposer une application de reconnaissance de gestes en temps réel qui consiste à détecter et à reconnaître le maximum des gestes d'une personne quelconque (par exemple, devant un distributeur automatique de billets et de prédire ses intentions (retrait d'argent ou acte de vandalisme) avec une base de données définie à partir des positions du corps humain.

Notre mémoire est organisé comme suit:

Dans le premier chapitre, nous présentons une vue d'ensemble de l'état de la recherche actuelle sur la vidéosurveillance, en commençant par un historique, ses différents composants, puis nous passons à la vidéosurveillance intelligente en mettant en évidence son importance, ses avantages et ses architectures. Nous terminons ce chapitre par les Domaines d'application des systèmes de vidéosurveillance.

Dans le deuxième chapitre, nous commençons par définir le geste, ses types et styles ainsi que ses fonctionnalités. Nous allons détailler les différentes étapes de la conception d'un système de reconnaissance de geste en donnant quelques exemples et nous le terminons par les domaines d'application.

Dans le troisième chapitre, nous avons présenté les Démarches adoptée pour concevoir le système de reconnaissance de gestes ou nous avons parlé des différentes étapes franchies pour la réalisation de notre système de reconnaissance de geste en commençant par la phase d'acquisition, puis la phase d'analyse et enfin la phase de la classification et d'interprétation.

Le quatrième chapitre présente l'implémentation pour la réalisation de notre système ainsi que toutes ses interfaces et résultats des tests effectués.

En fin, nous terminons ce mémoire par une conclusion générale qui résume, l'essentiel de notre travail, les contributions apportées et implémentées pour la conception de notre système, et bien sûr les perspectives envisagées pour son amélioration.

Chapitre I : Vidéosurveillance et vidéosurveillance intelligente

1. INTRODUCTION

La surveillance, qui était autrefois accomplie uniquement par des agents de sécurité, a vu l'arrivée de la technologie comme une révolution. La vidéosurveillance permet aux agents d'être plus efficaces en visualisant directement de leur poste de sécurité tous les points clés des locaux à surveiller.

Bien qu'elles soient d'une grande aide pour les agents, elles restent insuffisantes, car l'humain ne peut rester devant des écrans sans que rien ne lui échappe.

Par ailleurs, l'émergence des systèmes de vidéosurveillance intelligente offre la promesse d'améliorer la surveillance vidéo, à travers des logiciels pour analyser les flux vidéo provenant des caméras de surveillance.

Dans ce chapitre, nous présenterons les systèmes de vidéosurveillance, où nous décrirons leurs différents composants, ainsi que leurs évolutions dans le temps et nous terminerons par introduire les systèmes de vidéosurveillance intelligente.

2. Les systèmes de vidéosurveillance

2.1- Définition

La vidéosurveillance ou CCTV (Closed-Circuit TeleVision) est un système conçu pour surveiller des endroits publics ou privés à distance, à l'aide de caméras qui transmettent les images saisies à un équipement de contrôle qui les enregistre ou les reproduit sur un écran de contrôle. Elle a pour but de surveiller des lieux afin de garantir leurs sûretés, anticiper et prévenir les agressions, les vols et les différents crimes, ainsi que pour gérer les incidents et mouvements de foule.

On peut définir le système de vidéosurveillance (voir Fig.I.1) comme étant un système de visualisation en temps réel d'un environnement spécifique dont sa priorité est de maintenir l'accès à une zone protégée sous contrôle, prévenir l'intrusion et observer des secteurs publics pour détecter des comportements soupçonneux [1].

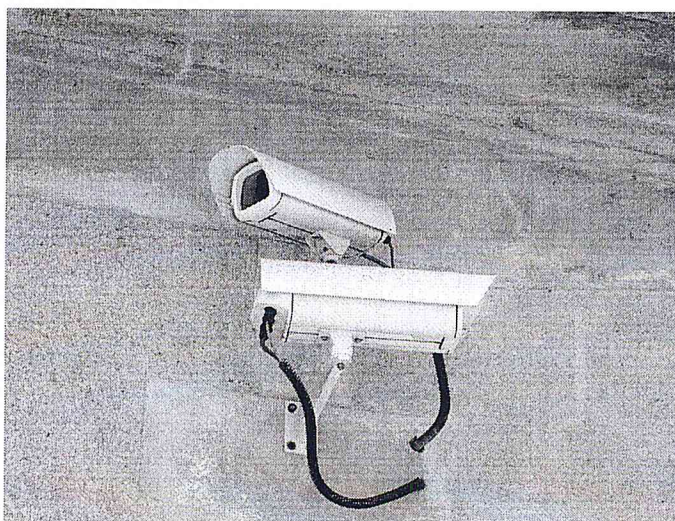


Fig.I.1: Caméras de surveillance.

2.2- Apparition

La vidéosurveillance fait partie de la famille des systèmes de sécurité électronique. Elle a vu le jour peu après la création de la télévision (dans les années 40). Le premier système fut mis au point en Allemagne, en 1942, par la compagnie SIEMENS. Avec cette invention, il était désormais possible de visualiser des images captées par une caméra (en noir et blanc à l'époque) et de les transmettre par câble sur un moniteur [2].

Les principes de base de ce que l'on appellera plus tard la CCTV étaient nés, mais elle s'est véritablement développée à partir des années 1970, principalement au Royaume-Uni pour lutter contre les activités terroristes de l'IRA (Irish Republican Army) [2].

Dans les années 1990, l'implantation de la vidéosurveillance s'est accrue par une série d'installations de caméras de la part du gouvernement. À l'heure actuelle, les caméras au Royaume-Uni couvrent la plupart des centres villes, et de nombreuses gares et parkings. Plus tard d'autres pays se sont intéressés à la vidéosurveillance telle que la France, où le nombre de caméras installées était estimé à un million en 1998 et sont présentes dans divers lieux tels que les aéroports, les gares, les routes et les transports publics [2].

2. 3- Évolution des systèmes de vidéosurveillance

Les systèmes de vidéosurveillance existent depuis environ 25 ans. Intégralement analogiques à leurs débuts, ils ont évolué progressivement vers la technologie numérique. Les systèmes actuels ne ressemblent guère aux anciennes caméras analogiques branchées sur des magnétoscopes traditionnels. Aujourd'hui, ils utilisent les caméras réseau et les serveurs informatiques pour l'enregistrement vidéo dans un système entièrement numérique [3].

Depuis leur apparition, les systèmes de vidéosurveillance n'ont pas cessé d'évoluer à travers les générations qui sont distinguées par :

➤ Système de Vidéosurveillance analogique avec magnétoscopes traditionnels

Un système de vidéosurveillance analogique utilise une caméra vidéo analogique avec sortie coaxiale qui la relie à un moniteur analogique (voir Fig.I.2), lequel restitue l'image à partir du signal vidéo transmis par la caméra. Pour enregistrer ces images vidéo, on utilise un magnétoscope traditionnel [4].

Dans le cas de systèmes analogiques d'envergure, un quad ou multiplexeur peut être connecté entre la caméra et le moniteur. Ce quad/multiplexeur permet alors de visualiser et d'enregistrer le contenu de plusieurs caméras sur un même moniteur et magnétoscope, mais selon une fréquence d'image cependant inférieure [4].

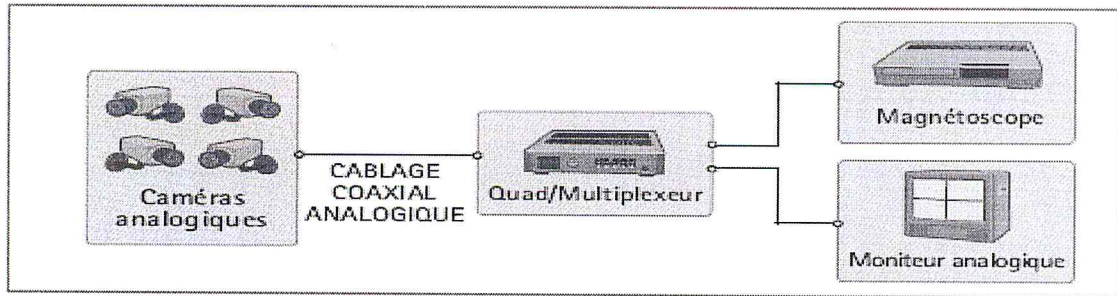


Fig.I.2: Schéma de vidéosurveillance analogique avec magnétoscope traditionnels [4].

➤ **Système de vidéosurveillance analogique avec enregistreur numérique**

Un système de vidéosurveillance analogique peut être couplé à un enregistreur numérique ou « digital » (voir Fig.I.3) [4].

Avec ce dispositif, l'enregistrement vidéo ne se fait plus sur bandes magnétiques mais sur des disques durs où les séquences vidéo sont numérisées et compressées de manière à emmagasiner quotidiennement un gros volume d'images. Par ailleurs, la plupart des enregistreurs numériques disposent de plusieurs entrées vidéo, ce qui leur permet d'intégrer d'emblée les fonctionnalités du quad ou des multiplexeurs [4].

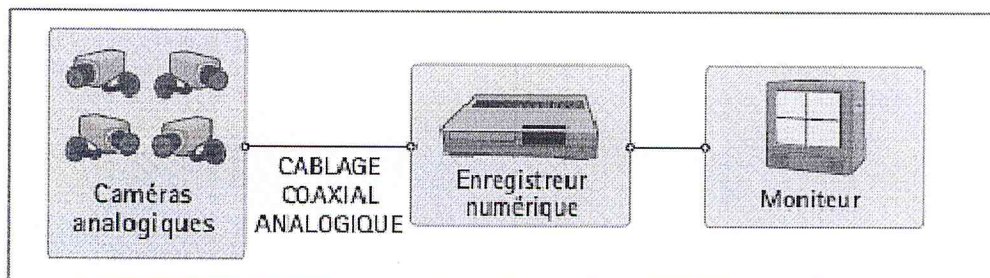


Fig.I.3: Schéma de vidéosurveillance analogique avec enregistreur numérique [4].

➤ **Système de vidéosurveillance analogique avec enregistreur numérique réseau**

Il est possible de configurer un système de vidéosurveillance analogique passant par un enregistreur numérique sur réseau grâce à une connexion via un port Ethernet (voir Fig.I.4) [4].

La vidéo étant numérisée et compressée sur l'enregistreur numérique, les images peuvent ainsi être transportées sur un réseau informatique pour visualisation sur PC distant. Sur certains systèmes, la restitution des images vidéo requiert un client Windows spécifique. Sur d'autres, il suffit d'un navigateur web standard, plus flexible pour une visualisation à distance. Les enregistreurs numériques sur réseau permettent donc la visualisation d'images vidéos à distance sur PC et le contrôle du système (de prise de vue/de la caméra) à distance [4].

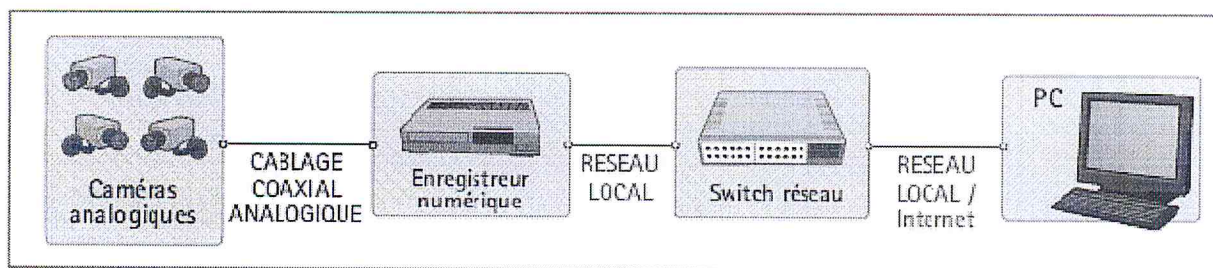


Fig.I.4: Schéma de vidéosurveillance analogique avec enregistreur numérique réseau [4].

➤ **Système de vidéosurveillance sur IP avec serveur vidéo**

La plupart des nouveaux systèmes de vidéosurveillance fonctionnent sur réseaux IP (voir Fig.I.5). Associés à un serveur vidéo, ils comprennent une caméra analogique, un serveur vidéo, un commutateur réseau et un PC équipé d'outils de gestion vidéo. Le serveur vidéo sert de pierre angulaire : il assure la migration du système de vidéosurveillance analogique vers une solution de vidéosurveillance sur IP. Ainsi, la caméra analogique est branchée sur le serveur vidéo qui assure la numérisation et la compression des séquences vidéo. Le serveur vidéo est connecté sur le réseau informatique qui transporte les images vidéo vers un PC ou un serveur via un commutateur réseau. Les images vidéo sont alors enregistrées sur le disque dur du PC [4].

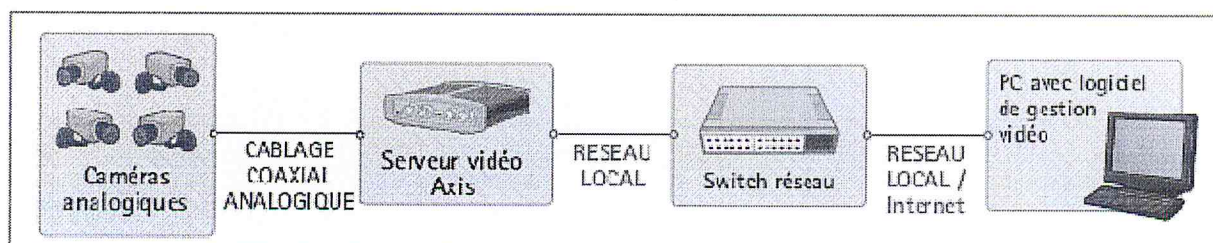


Fig.I.5: Schéma de vidéosurveillance sur IP avec serveur vidéo[4].

➤ **Système de vidéosurveillance sur IP avec caméras réseau**

Il s'agit d'un système entièrement numérique dans lequel les images vidéo sont transmises sur réseau IP à l'aide de caméras réseau (voir Fig.I.6). Une caméra réseau associe une caméra numérique et un ordinateur, et permet la numérisation et la compression vidéo. Les images vidéo sont acheminées par réseau IP via les commutateurs réseaux, pour être restituées et enregistrées sur un PC/serveur standard à l'aide d'outils de gestion vidéo. Ce nouveau système de vidéosurveillance sur IP « tout numérique » permet une meilleure flexibilité et une bonne évolutivité [4].

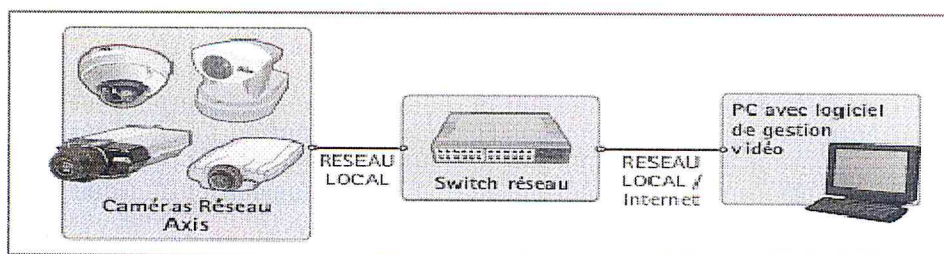


Fig.I.6 : Schéma de vidéosurveillance sur IP avec caméras réseau[4].

3. Éléments constituant un système de vidéosurveillance

Nous allons présenter ci-dessous les différents accessoires qui peuvent composer un système de vidéosurveillance.

3.1- Caméra

La caméra est un système de prise d'images animées qui génère un signal vidéo en noir et blanc ou en couleur. Elle capte la lumière pour la transformer en signal électrique. Elle se compose parfois d'un capteur d'image et d'un outil de traitement permettant de générer le signal vidéo [5].

Nous pouvons trouver actuellement sur le marché un afflux de caméras. Nous allons énumérer et présenter quelques-unes, notamment celles que l'on trouve le plus couramment dans le domaine de la vidéosurveillance.

➤ Caméras infrarouges (IR) :

Ce que l'homme ne peut pas distinguer dans l'obscurité, la caméra IR le voit. Grâce aux LED qui entourent l'objectif de la caméra IR, vous bénéficierez d'une parfaite retransmission d'images, jusqu'à 100 mètres, même dans les zones les plus obscures [6].

➤ Caméra discrète :

Une caméra cachée est une caméra placée de manière à pouvoir filmer des individus à leur insu. Il peut y avoir plusieurs caméras braquées sur le sujet à piéger, filmant à différentes distances et de différents points de vue [5].

➤ Caméra analogique :

Ces caméras sont facilement reconnaissables ; elles ont une sortie de type BNC (Bayonet Neil Concelman). La liaison se fait via le câble coaxial, leurs seuls rôles est de capturer les images et les envoyer à un enregistreur [5].

3. 2- Caméra IP

Une caméra IP ou caméra réseau (Fig.I.7), peut être représentée comme la jonction en une seule unité d'une caméra et d'un ordinateur. Elle capture et envoie des images en temps réel à travers un réseau IP, permettant aux utilisateurs autorisés de visualiser localement ou à distance, stocker et gérer la vidéo à travers une infrastructure de réseau IP [5].

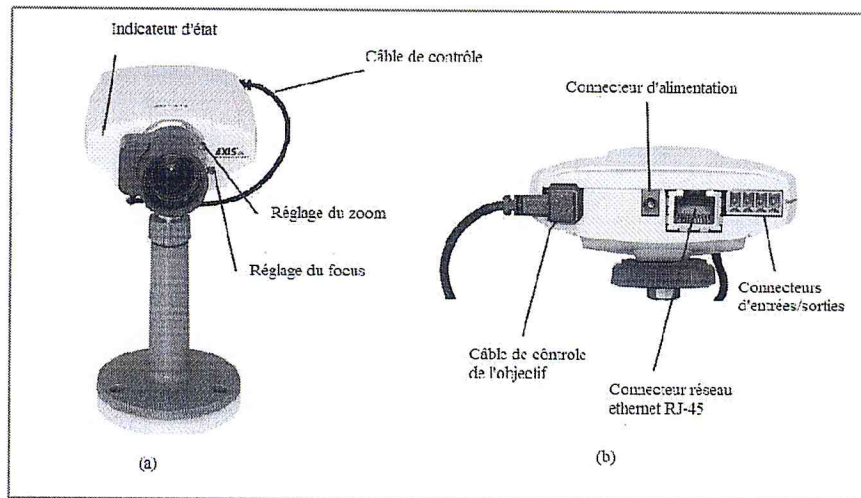


Fig.I.7 : Caméra IP. (a) : face avant (b) : face arrière

3. 3- Enregistreur vidéo

Dans un système de vidéosurveillance, la majeure partie de la vidéo n'est pas visualisée en temps réel, car l'opérateur humain ne peut pas surveiller toutes les caméras en même temps, d'où la nécessité de sauvegarder la vidéo. Afin d'accomplir cette tâche, un enregistreur vidéo doit être utilisé. Nous distinguons deux types d'enregistreur vidéo, le VCR (Video Cassette Recorder) et le DVR (Digital Video Recorder) :

➤ Le VCR (Video cassette Recorder) :

Le VCR enregistre un signal vidéo analogique sur une bande magnétique avec une capacité allant de 6 à 24 heures selon le mode d'enregistrement. Afin d'enregistrer le signal provenant de plusieurs caméras, un multiplexeur doit être utilisé, car le VCR possède une seule entrée vidéo. La durée nécessaire pour localiser une image particulière dans la bande magnétique peut atteindre plusieurs minutes, ce qui rend la recherche d'événements dans ces enregistrements très pénible [5].

➤ **Le DVR (Digital Video Recorder) :**

Le DVR enregistre un signal vidéo numérique sur un disque dur et la vidéo est compressée avant l'enregistrement, ce qui le rend largement meilleur que le VCR en termes de capacité de stockage. De plus, le DVR possède jusqu'à 16 ports d'entrées lui permettant ainsi d'enregistrer la vidéo provenant de plusieurs caméras en même temps, et un accès direct à n'importe quelle image enregistrée sur le disque en un temps record [5].

3. 4- Encodeur vidéo

Un encodeur vidéo (voir Fig.I.8) aussi appelé serveur vidéo, est un dispositif permettant l'intégration des caméras analogiques dans un système de vidéosurveillance numérique. Il possède des ports d'entrées analogiques pour connecter les caméras analogiques, une unité de traitement qui sert à numériser et à compresser la vidéo analogique, et un port Ethernet pour envoyer la vidéo sur le réseau pour un éventuel traitement ou enregistrement sur un serveur standard ou un PC. Un encodeur vidéo peut également supporter d'autres fonctionnalités telles que l'audio et l'activation d'alarme via des entrées et sorties numériques [5].

3. 5- Décodeur vidéo

Comme le montre la figure (Fig.I.8), dans certaines installations, la vidéo et le flux audio circulant dans le réseau doivent être visualisés sur un équipement analogique. En utilisant un décodeur vidéo, la vidéo et le flux audio provenant du réseau sont convertis en signaux analogiques, et peuvent alors être reliés aux moniteurs analogiques. Un décodeur vidéo peut aussi être utilisé avec un encodeur vidéo pour le transport de la vidéo sur de longues distances, à partir d'une caméra analogique à un moniteur analogique [5].

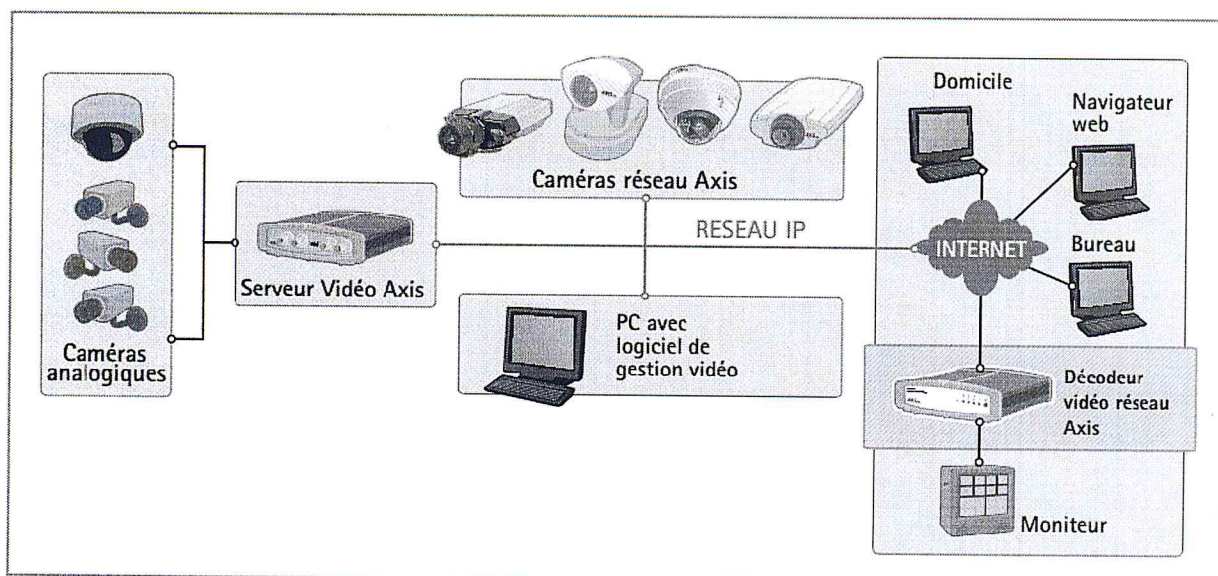


Fig.I.8 : Schéma illustrant l'utilisation des encodeurs et décodeurs vidéo dans un système de vidéosurveillance [5].

4. Composantes fonctionnel d'un système de vidéosurveillance

Dans cette section nous allons présenter, de façon sommaire, les différentes tâches ou fonctions d'un système de vidéosurveillance :

➤ Acquisition :

Il s'agit des différents modèles de caméras, que ce soit analogique ou numérique utilisées pour acquérir les vidéos / sons dans les systèmes de surveillance. Avant l'arrivée des caméras IP l'enregistrement de la vidéo se faisait sur des bandes magnétiques, cela a ensuite été remplacé par des enregistreurs numériques (DVR) qui convertissent les vidéos analogiques en numériques. Les systèmes de vidéosurveillance sur IP utilisent des caméras acceptant la transmission des données numériques sur un réseau Ethernet ou un réseau sans fil en utilisant le protocole IP [7] [5].

➤ Transmission :

La vidéo capturée par la caméra doit être transmise sur le réseau afin de la visualiser, l'enregistrer et éventuellement effectuer certains traitements. La transmission est effectuée en utilisant l'un des moyens suivants [7] [5] :

- Par câbles (câbles coaxiaux, câbles à fibres optiques, fils de cuivre, RJ45).
- Sans fil (infrarouge, fréquences radio).

La transmission par câble offre une large bande passante et une meilleure fiabilité par rapport à la transmission sans fil et ceci à un coût réduit. Cependant, la transmission sans fil peut s'avérer plus intéressante dans le cas d'un périmètre de surveillance large ou quand le secteur sous surveillance ne peut pas être atteint par des câbles.

➤ **Compression :**

Le traitement des signaux numériques est plus facile par rapport aux signaux analogiques. Les méthodes de compression incluent des techniques de suppression des portions non pertinentes d'images et les portions redondantes du flux media. La compression implique la réduction de la taille des données par un processus d'élimination des informations non pertinentes [7] [5].

Il y'a deux principales raisons pour lesquelles la vidéo doit être compressée :

- Réduire la quantité de stockage utilisée pour la vidéo de façon à ce que les disques durs puissent stocker beaucoup plus de vidéos.
- Réduire la bande passante nécessaire pour la transmission de la vidéo sur le réseau.

➤ **Traitement :**

Le traitement d'images est effectué par un système de gestion de vidéo. Ce système inclus la visualisation, l'enregistrement, l'analyse et la recherche dans les séquences vidéo enregistrées. Il existe quatre catégories de systèmes de gestion de la vidéo [7] [5] :

- Enregistreur vidéo numérique (DVR) : il contient un disque dur interne pour le stockage de la vidéo numérique, une partie application intégrée pour le traitement de la vidéo. Son rôle consiste uniquement à prendre des vidéos provenant des caméras analogiques et à les numériser.
- Enregistreur vidéo numérique hybride (HDVR) : il est similaire au DVR sauf qu'il accepte aussi la vidéo provenant des caméras IP.
- Enregistreur vidéo réseau (NVR) : il est conçu pour les systèmes de vidéosurveillance sur IP. Il n'accepte que la vidéo provenant des caméras IP ou des DVRs.
- Logiciel de vidéosurveillance sur IP : Ce système contient qu'une partie logicielle et pas de partie matérielle. Il est conçu pour les systèmes de vidéosurveillance sur IP. Un explorateur web peut être utilisé pour la gestion de quelques caméras, mais dans le cas

d'un système de vidéosurveillance avec un nombre important de caméras, un logiciel spécifique doit être conçu pour gérer ce système.

➤ **Archivage :**

Dans un système de vidéosurveillance typique, certains critères doivent être pris en compte pour le stockage, tels que le nombre de caméras, le temps d'enregistrement, le mode d'enregistrement (permanent ou déclenché par événement), la qualité d'image et le nombre d'images par seconde [7] [5].

Il existe deux types de dispositif de stockage [7] [5]:

- Buffer interne : les caméras IP contiennent une mémoire intégrée de type flash ou DRAM (Dynamic Random-Access Memory) pour permettre le stockage temporaire des images. Ce type de stockage est adapté aux systèmes ayant moins de 50 caméras.
- Network Attached Storage (NAS) : offre un espace de stockage partagé entre les différents clients du réseau. Ce type est mieux approprié aux systèmes de grande taille, il garantit l'évolutivité, la flexibilité et la redondance.

Afin de minimiser le risque de perte de données en cas d'endommagement d'un disque dur, les disques RAID (RedundantArray of Independent Disks) sont utilisés pour pallier ce problème.

➤ **Visualisation :**

La majeure partie des vidéos provenant des caméras d'un système de vidéosurveillance ne sont pas visualisées, elles sont directement stockées en cas d'incident ou pour une éventuelle recherche plus tard. Les grands centres de sécurité utilisent une rangée de grands écrans pour visualiser quelques vidéos, qui pourraient également être la commutation entre plusieurs caméras [7] [5].

5. Les systèmes de vidéosurveillance intelligente

5.1- Définition

La vidéosurveillance intelligente est une technologie qui permet, en utilisant un logiciel, d'extraire automatiquement dans des séquences vidéos les différents objets pertinents et leurs détails. Elle transforme la vidéo en données qui seront transmises ou archivées pour permettre au système de vidéosurveillance (voir Fig.I.9) de déclencher une action selon la donnée, par

exemple: activer des appareils de surveillances supplémentaires, donner une alerte ou verrouiller des portes[5].

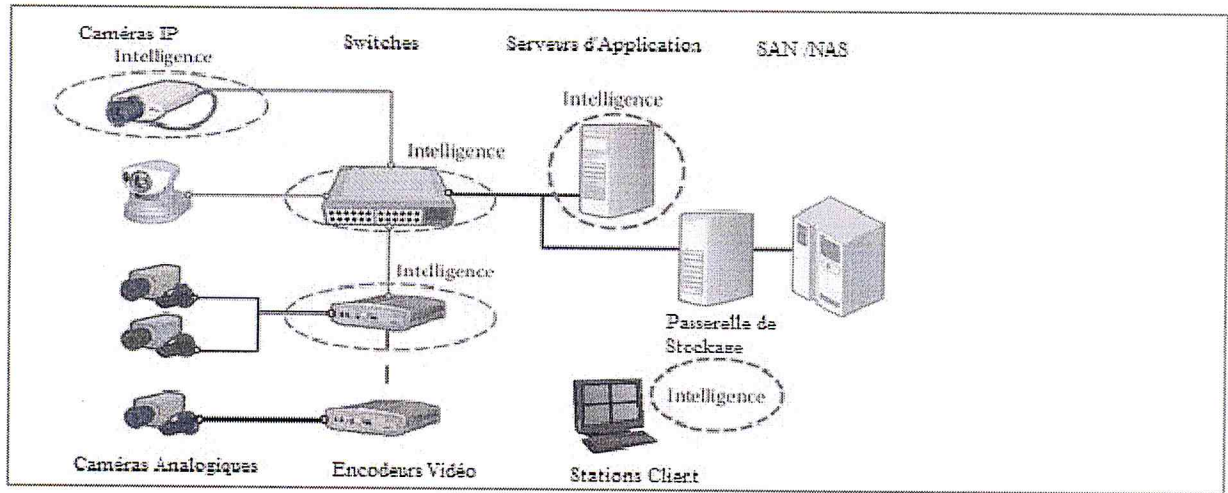


Fig.I.9 : Schéma illustrant les différentes parties d'un système de vidéosurveillance qui peuvent être équipées des fonctionnalités d'intelligence [5].

5. 2- L'importance de la vidéosurveillance intelligente

Une des raisons les plus importantes au besoin d'utiliser un système de vidéosurveillance intelligente est la grande quantité d'informations venant d'un nombre considérable de caméra dans les systèmes de vidéosurveillance. Le suivi et l'analyse de toutes ces vidéos pour détecter les comportements ou événements suspects devient difficile pour un agent de sécurité, et la probabilité pour que l'agent repère un événement pour réagir devient faible. Surtout si le personnel des centres de surveillance est requis pour d'autres tâches telles que le contrôle d'accès, le contrôle des communications radios, etc. Ce qui rend la vidéosurveillance intelligente une bonne solution à ce problème [7] [5].

5. 3- Avantages de la vidéosurveillance intelligente

Parmi les avantages de la vidéosurveillance intelligente [7] [5]:

- Elle est fonctionnelle 24/24 et 7/7.
- Elle permet une intervention en temps réel, en enclenchant une alarme pour informer l'opérateur ou en commandant le déplacement ou le zoom d'une caméra pour une surveillance plus précise de l'événement.
- Réduction de la bande due à la transmission ou l'enregistrement que des données contenant des événements pertinents.

- Elle dispense les agents de sécurité d'une surveillance permanente
- Recherche rapide d'événements pertinents dans les séquences vidéo archivées.
- Identification automatique des objets dans une scène et suivi de leur activité.

5. 4- Architectures des systèmes de vidéosurveillance intelligente

Il y a deux catégories d'architectures pour mettre en place un système de vidéosurveillance intelligente : centralisée et distribuée.

Dans les architectures centralisées, la vidéo et d'autres informations sont collectées par les caméras, puis elles sont acheminées vers un serveur centralisé pour l'analyse. Dans les architectures distribuées, les caméras réseaux ou les encodeurs vidéo, ou d'autres composants du réseau (par exemple, commutateurs) sont « intelligents » et sont capables de traiter la vidéo et d'extraire les informations pertinentes [7] [5] [4].

5. 4-1 Architecture centralisée

Dans les architectures centralisées, toute la vidéo obtenue par les caméras est acheminée au serveur pour un traitement centralisé. Les infrastructures basées sur des caméras analogiques emploient la plupart du temps des DVRs comme serveur, tandis que dans un système utilisant des caméras réseau, des PC-serveurs sont utilisés pour le traitement automatique [7] [5] [4].

➤ Architecture centralisée à base de DVR

Dans les systèmes CCTV traditionnels de télévision, la vidéo obtenue par les caméras analogiques est introduite dans un DVR équipé des fonctionnalités de vidéo intelligente (voir Fig.I.10). Les DVRs ont des encodeurs qui convertissent la vidéo du format analogique au numérique, puis ils exécutent l'analyse intelligente (par exemple, le comptage des gens ou l'identification des numéros d'immatriculation des véhicules). Ils compressent également la vidéo, l'enregistrent, et distribuent les alarmes et la sortie vidéo aux opérateurs autorisés [7] [5] [4].

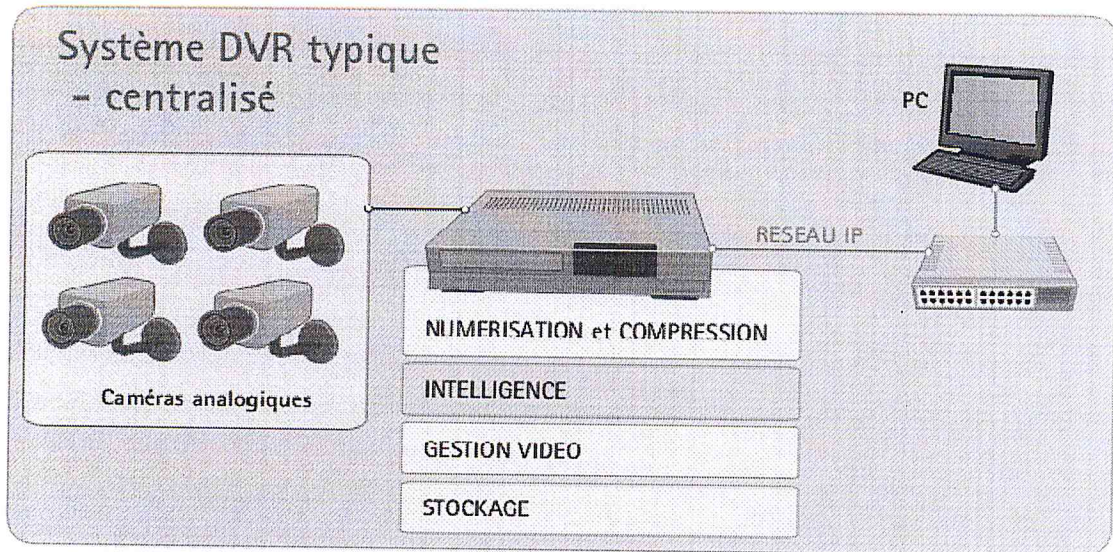


Fig.I.10: Système centralisé à base de DVR (Les fonctionnalités d'intelligence se situent dans le DVR) [4].

➤ **Architecture centralisée à base d'un PC-server**

Pour surmonter les insuffisances des DVRs, de nouvelles architectures centralisées ont été conçues, elles utilisent un PC-serveur pour le traitement de la vidéo (voir Fig.I.11). La vidéo provenant des caméras réseaux est directement acheminée au serveur, tandis que la vidéo provenant des caméras analogiques doit être numérisée d'abord par un encodeur vidéo, puis acheminée vers le serveur [7] [5] [4].

Cette architecture est plus flexible et extensible que celle à base de DVR parce que la numérisation et la compression ne se font plus par le système de gestion de la vidéo, mais plutôt par les caméras réseaux ou les encodeurs vidéo dans le cas des caméras analogiques, ceci permet d'alléger les tâches de système de gestion de la vidéo [7] [5] [4].

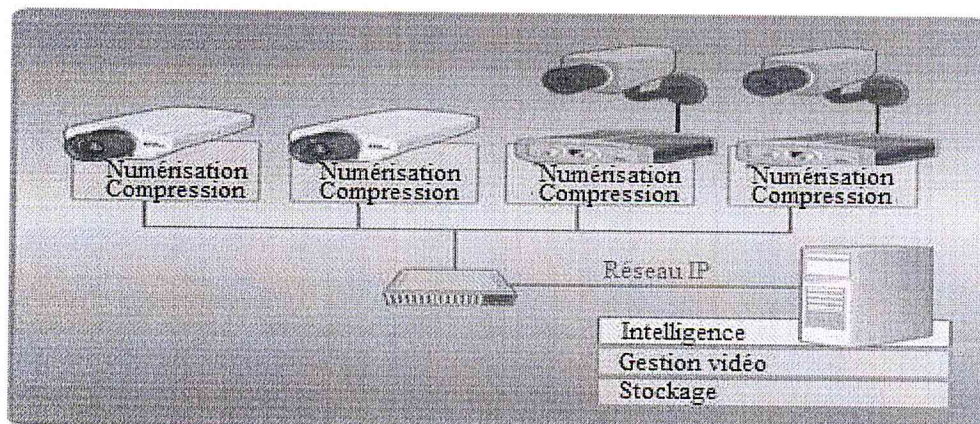


Fig.I.11: Système centralisé à base d'un PC serveur (Les fonctionnalités d'intelligence sont dans le serveur).

5. 4-2 Architecture distribuée

Les architectures distribuées (voir Fig.I.12) sont conçues pour surmonter les limitations des systèmes centralisés qui surchargent le point central du système, tel qu'un PC serveur ou un DVR. En distribuant le traitement à différents éléments dans un réseau, la consommation de la bande passante peut être réduite, mais aussi, les calculs nécessaires à l'analyse vidéo pourront être effectués sur des caméras intelligentes. De plus, ces architectures sont souples, puisque l'ajout de caméras n'est pas forcément limité par la puissance de calcul de l'enregistreur numérique ou du serveur [4] [5] [7].

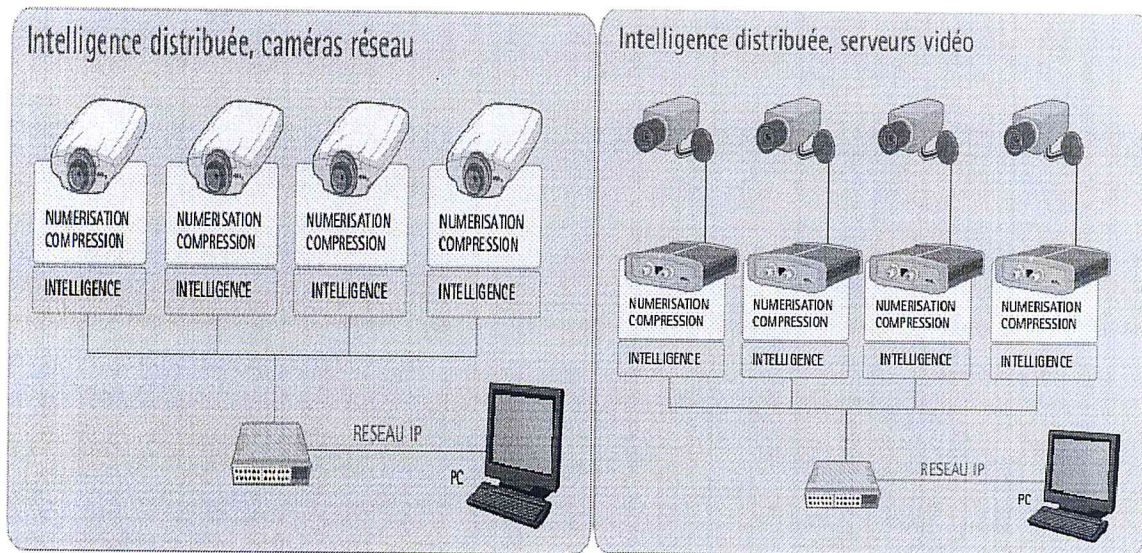


Fig.I.12: Système distribué, les fonctionnalités d'intelligences sont aux extrémités de réseau (à gauche cas de caméras IP, à droite cas de caméras analogiques)[4].

6. Domaines d'application des systèmes vidéosurveillance

La vidéosurveillance est devenu très importante, c'est pourquoi elle est utilisée dans plusieurs domaines, tel que [7] [5] :

➤ Commerce

Le domaine du commerce est un marché croissant pour la vidéosurveillance intelligente. Celle-ci est utilisée soit pour la sécurité interne (magasin et entrepôt), soit pour la sécurité externe (parkings). Dans les plus petits magasins, les caméras sont utilisées pour prévenir en cas de vol ou d'incident. Dans les grands magasins, les systèmes de vidéosurveillance, beaucoup plus sophistiqués, sont installés pour la surveillance des différents endroits.

➤ **Transport**

La sécurité et le bon fonctionnement des stations de bus, les ports, et les transports en commun sont des points critiques dans l'économie d'un pays. Cependant, le secteur de transport pose des problèmes techniques majeurs aux systèmes de vidéosurveillance intelligente à savoir le nombre de personnes passant dans le champ d'une caméra, la diversité des comportements des passagers et les conditions défavorables pour les caméras (les vibrations, la poussière, etc.).

Les logiciels de traitement ciblent le secteur de transport en lui offrant les différentes fonctions adaptées telles que :

- La détection d'intrusion dans un périmètre contrôlé.
- Reconnaissance faciale.
- Reconnaissance des plaques d'immatriculation pour la surveillance d'accès au parking.
- Détection des comportements suspects.

➤ **Les banques :**

La vidéosurveillance est couramment employée pour la sécurité des banques. La présence des caméras agit d'abord en tant que moyen de dissuasion au vol à main armée, et les vidéos de la surveillance aident aussi à détecter la fraude telle que l'installation des dispositifs pour lire les informations magnétiques sur les cartes bancaires.

7. Conclusion

La vidéosurveillance ne date pas d'aujourd'hui. Analogique à ses débuts, elle est de nos jours de plus en plus numérisée. L'avènement des réseaux IP (Internet Protocol) à haut débit et l'utilisation du logiciel dans la vidéosurveillance intelligente ouvrent la voie à une quantité d'applications innovantes et très performantes.

L'industrie de la vidéosurveillance englobe aujourd'hui toute une variété de systèmes et d'équipements, le développement de ces systèmes a poussé les chercheurs du domaine de la vidéo surveillance à trouver de nouvelles innovations, telles que les systèmes de reconnaissance de gestes avec la nouvelle technologie de capteur de vision Kinect afin d'augmenter la performance de la vidéo protection. Pour mieux comprendre cette technologie le chapitre suivant sera consacré au système de reconnaissance de geste.

Chapitre II: La reconnaissance de gestes

1. INTRODUCTION

L'intérêt progressif que porte la communauté des chercheurs en vision artificielle à l'analyse du mouvement humain se manifeste dans une très grande variété de contextes : de l'analyse de performance en sport à la surveillance, en passant par les interfaces homme-machine, et de la réalité virtuelle à la réalité augmentée [8].

L'évolution des capteurs et la variation de leurs domaines d'utilisation ont permis l'augmentation de l'importance de la reconnaissance des gestes par vision artificielle.

En raison de la diversité des mouvements du corps humain, l'interprétation des gestes par vision artificielle peut être difficile car leurs apparences dépendent du point d'observation, de la trajectoire des parties concernées par le mouvement du corps, en plus des difficultés classiques en vision, notamment la variation de luminosité, contraste, les obstacles, le bord du cadre du capteur.

La reconnaissance de gestes est une tâche complexe impliquant divers aspects dont la modélisation et l'analyse de mouvement, la reconnaissance des formes et les méthodes d'apprentissage, voir même les études psycholinguistiques.

Plusieurs approches ont montré leurs efficacités pour la reconnaissance et la classification des mouvements humains. Certaines procèdent par la détection et le suivi des trajectoires du corps entier ou des parties du corps, d'autres par l'analyse géométrique et temporelle de ces dernières et l'extraction des caractéristiques de leur apparence dans les images.

Dans ce chapitre nous allons voir ce que c'est un geste et ses différentes caractéristiques, la reconnaissance des gestes, le système de reconnaissance et ses étapes (où nous verrons les outils d'acquisition et les méthodes de classifications) et à la fin nous montrons quelques exemples.

2. Définition du geste :

Le geste est souvent assimilé à un mouvement du corps ou une partie de lui, en particulier les mains, les bras ou la tête que l'on fait avec ou sans intention de signifier quelque chose.

Les gestes sont des moyens de communication importants que l'être humain possède. Ils permettent d'interagir avec le monde extérieur pour réaliser différentes tâches [9].

Le geste peut être considéré en tant qu'une action physique (il est le résultat des extensions ou des rétractions musculaires soit volontaires ou réflexes) ou une action interactive (un geste pour réaliser une communication non verbale) [9].

Un des problèmes dans le domaine de la recherche sur les gestes est le manque des termes qui décrivent les interactions, c'est pourquoi plusieurs taxonomies sont proposées selon, par exemple, le type du geste, le domaine d'applications ou selon les technologies disponibles et utilisées. Trois classifications sont présentées pour les gestes en fonction des parties du corps humain [9] :

- **Les gestes de la main et du bras** : ils forment la principale catégorie des gestes interactifs. La main permet de réaliser des gestes précis et complexes. Les recherches autour de ces gestes concernent principalement la reconnaissance de positions de la main, l'interprétation du langage des signes et le développement d'interface homme-machine permettant la manipulation et l'interaction avec des données ou des éléments d'un environnement virtuel.
- **Les gestes de la tête et du visage** : peu de gestes de la tête ont une signification spécifique; l'orientation de la tête est quant à elle très utile pour la détection des champs de vision. Les recherches dans ce domaine s'intéressent à la reconnaissance faciale comme moyen d'authentification biométrique, comme soutien à d'autres systèmes de reconnaissance tels que la reconnaissance de la parole ; l'analyse des gestes faciaux est également utile pour la réalisation d'avatars virtuels réalistes ou encore pour décrypter les émotions à des fins marketings ; estimation de l'âge ou recherche des suspects à travers les similarités faciales.
- **Les gestes impliquant tout le corps** : les recherches dans ce domaine s'intéressent à tout le corps en interaction avec son environnement (par exemple, l'analyse des gestes d'un danseur afin de générer de la musique appropriée ; analyse des gestes d'un athlète pour améliorer ses performances). Nous différencions également les gestes dynamiques des gestes statiques. Un geste statique, est l'information que porte le corps à un moment fixe, qui est

exprimée par une seule posture alors que le geste dynamique désigne une succession continue de postures, ce type est plus difficile à traiter.

3. Types de Geste :

Les gestes émis par un opérateur humain pour communiquer avec une machine sont de différentes natures et peuvent jouer différents rôles [10] :

- Geste instrumental (faire en montrant, manipuler des objets virtuels).
- Geste de désignation, de pointage.
- Configuration statique référençant une commande isolée.
- Geste de commande, désignant une action et ses paramètres (quoi et comment).
- Geste Co-verbal accompagnant une communication parlée.
- Geste fait de manière inconsciente.
- Geste fait de manière intentionnelle : dire et montrer comment.
- Gestes utilisés dans le langage des signes

4. Styles de gestes :

Dans [11], présentent cinq styles de gestes concernant la main et le bras, synthétisant les types d'interaction décrites à travers la littérature scientifique :

- Les gestes déictiques sont des gestes de pointage permettant d'identifier un objet ou son emplacement, et ils sont typiquement utilisés dans des environnements virtuels. Ces gestes peuvent être considérés comme implicites dans d'autres formes de gestes (par exemple lorsque l'on pointe un objet à manipuler). Le premier exemple fut le « Put that there » de R. A. Blot [12].
- Les gestes de manipulation dont le but est de contrôler une entité en appliquant une relation étroite entre le mouvement du geste et l'entité qui est manipulée.
- Les gestes sémaphoriques font partie de tout système gestuel basé sur un catalogue conventionnel de gestes statiques ou dynamiques (par exemple : le geste statique pour signifier « ok » ou le signe dynamique de la main pour dire « au revoir »).
- Les gesticulations sont les gestes les plus naturels et modèles de gestes Co-verbales. L'interprétation de ce type de gestes est le domaine de recherche en reconnaissance de geste le plus ambitieux car contrairement aux gestes sémaphoriques, leur signification

ne peut pas être issu directement d'un dictionnaire de gestes et doivent être mis en relation avec d'autres méthodes comme la parole.

– Le langage des signes : les gestes utilisés dans le langage des signes sont souvent considérés indépendamment des autres types de gestes étant donné qu'ils sont basés sur des principes de linguistique et qu'ils permettent de combiner gestes et signes pour former des structures grammaticales utiles à la conversation.

5. Fonctionnalités du geste :

A l'égard de la nature expressive du canal gestuel, nous distinguons trois fonctions du geste: le geste ergotique, le geste épistémique et le geste sémiotique, qui sont présentés comme suit :

- **Le geste ergotique (de la force/travail):**

Il s'agit d'appliquer une action sur le monde physique pour effectuer de différentes interactions (telles que la saisie, le transport, la modélisation, la transformation, la destruction...) [13].

- **Le geste épistémique (de connaissances):**

C'est les gestes de l'être humain pour explorer et acquérir des connaissances sur l'environnement (en utilisant essentiellement la main) comme le sens du toucher pour connaître les informations relatives à la forme, l'orientation, la distance, la taille, le comportement de l'objet...etc. [13].

- **Le geste sémiotique (sémantiques) :**

C'est les gestes qui permettent de performer des informations visuelles, comme le langage des signes, le geste Co-verbal (qui accompagne la parole) ou les gestes de communications basiques (gestes strictement physiques) utilisé dans des situations où on ne peut pas utiliser la parole (un environnement bruyant ou en plongée sous-marine, par exemple).

La fonction sémiotique du geste est celle qui est la plus riche et la plus complexe et elle est la plus exploitée pour les interactions homme-machine car elle se focalise sur l'aspect visuel des gestes pour bien transmettre leurs significations [13].

6. La reconnaissance de geste :

La reconnaissance de geste est un sujet informatique qui a comme but d'interpréter les gestes de l'humain via des algorithmes mathématiques, qui désigne l'ensemble des opérations permettant d'analyser une scène à savoir la capture des gestes (par exemple à l'aide d'une caméra ou d'un gant dotés de capteurs), la segmentation, l'évaluation des poses et l'interprétation à proprement parler.

Les gestes peuvent provenir de n'importe quel mouvement du corps ou d'une pose. Le domaine de l'interaction homme-machine et la variété et la complexité des gestes humains ont poussé les chercheurs à se focalisés sur la reconnaissance des gestes.

La reconnaissance des gestes peut être considérée comme un moyen pour les ordinateurs de commencer à comprendre le langage du corps humain, créant ainsi une liaison plus forte entre les machines et les humains, autre que celle offerte par les interfaces primitives en mode texte ou même des interfaces graphiques, qui limitent encore la communication au clavier et à la souris. En ce qui concerne la programmation de la reconnaissance du mouvement, celle-là peut être accomplie avec des techniques de vision par ordinateur et de traitement d'images [14].

Il y a un certain nombre d'avantages offerts par l'utilisation des gestes pour l'interaction :

- Une interaction naturelle : les gestes sont une forme naturelle de l'interaction, et sont faciles à utiliser ;
- Simple et puissant : un geste simple peut être utilisé pour spécifier à la fois une commande ainsi que ses paramètres ;
- Une interaction directe : la main utilisée en tant que périphérique d'entrée élimine le besoin d'objets intermédiaires entre la personne et la machine.

7. Système de reconnaissance de gestes :

Le processus de reconnaissance de gestes est proche du raisonnement de l'être humain, d'où s'inspirent les domaines de l'intelligence artificielle, qui servent à automatiser des tâches humaines. Le système de reconnaissance de geste est schématisé dans la figure ci-dessous (voir Fig.II.1), résumant les étapes qui seront détaillées par la suite pour l'identification du geste.

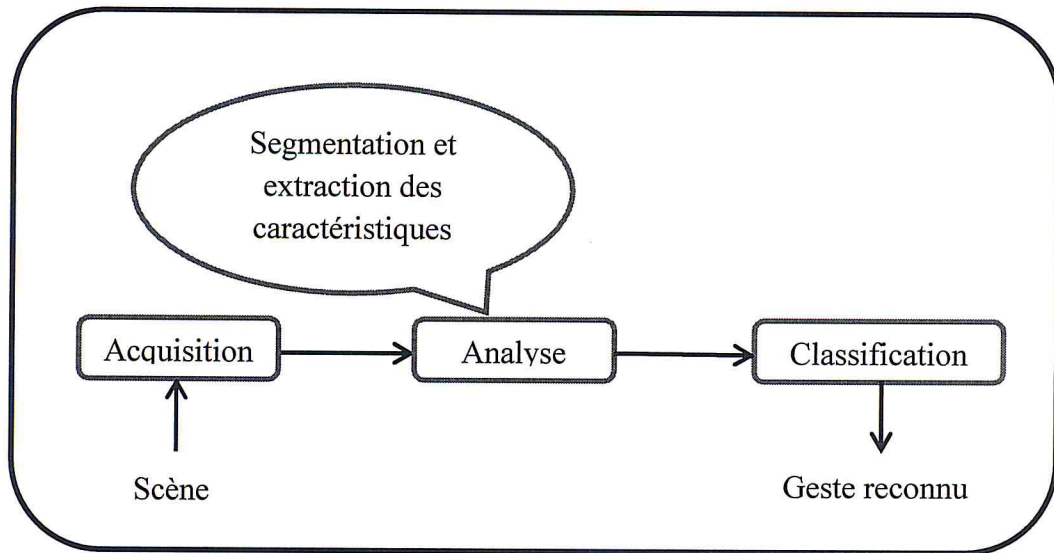


Fig.II.1: Un système de détection et de reconnaissance de Geste.

Voici le schéma d'un système de reconnaissance de geste (voir Fig.II.2) qui est présenté par Braffort et al.

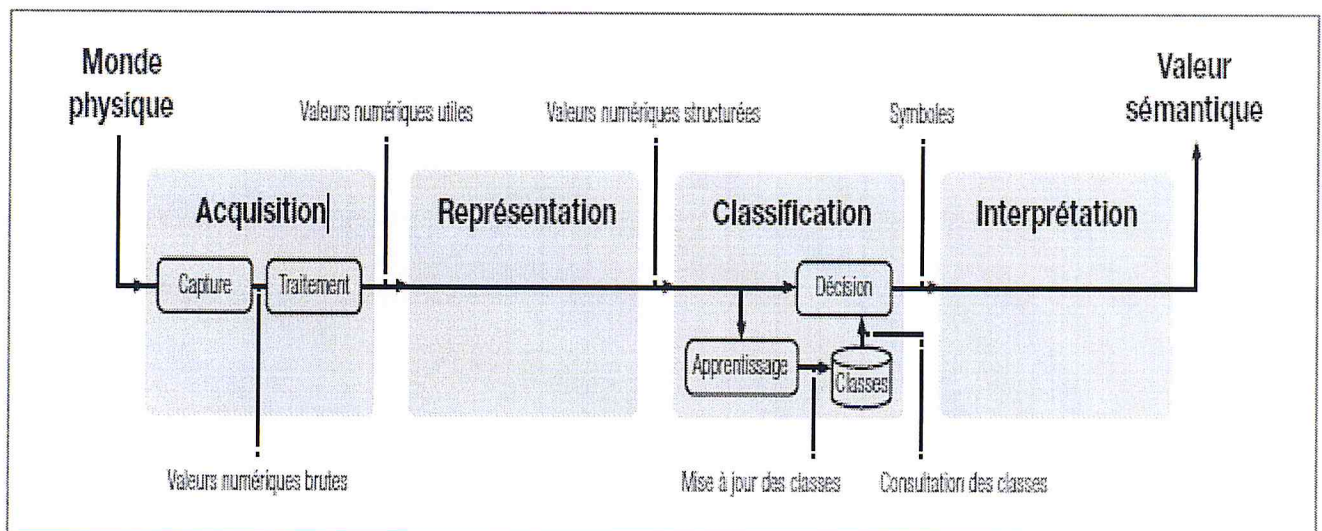


Fig.II.2: Schéma d'un système de reconnaissance de geste de Braffort et al. 2006[15].

7.1 Gestion d'un système d'apprentissage et de reconnaissance du geste pour l'interaction :

La plupart des systèmes de reconnaissance de gestes en informatique suivent quatre principales étapes. Elles se succèdent le plus souvent de façon séquentielle :

- L'acquisition de données, qui inclut la capture des données brutes et leur traitement éventuel;
- Leur représentation en valeurs numériques structurées (primitives);
- La classification qui permet de transformer le flux de données en symbole (classes de geste) qui constitue le vocabulaire de gestes de l'application;
- L'interprétation qui permet de traduire ces éléments symboliques en fonction du contexte.

Nous proposons de détailler ce processus dans les paragraphes suivants, en décrivant dans un premier temps les principes d'acquisition du geste et le matériel existant. Nous verrons ensuite comment décrire le geste en vue de sa reconnaissance par un système informatique.

Nous recensons quelques techniques d'apprentissage fréquemment utilisées pour la reconnaissance de gestes et enfin, la phase d'interprétation.

7. 1-1 Acquisition du geste :

L'interaction par le geste est possible grâce aux systèmes de capture de mouvement du corps. La modélisation des différentes parties du corps peut être très précise, en temps-réel, avec des données transmises à une fréquence élevée (> 60 Hz) [15].

Cependant, les systèmes de capture de mouvement restent complexes à mettre en place et à calibrer. Aussi, ils restent coûteux et nécessitent le plus souvent de grands espaces. Ils demeurent donc difficilement utilisables par le grand public [15].

Nous ciblons un interfaçage accessible qui soit à la fois peu intrusif et abordable financièrement. Ainsi, une option possible est de se tourner vers des dispositifs grand public principalement issus du domaine des jeux-vidéo. Nous proposons une liste non exhaustive des principaux périphériques (voir Fig.II.3).

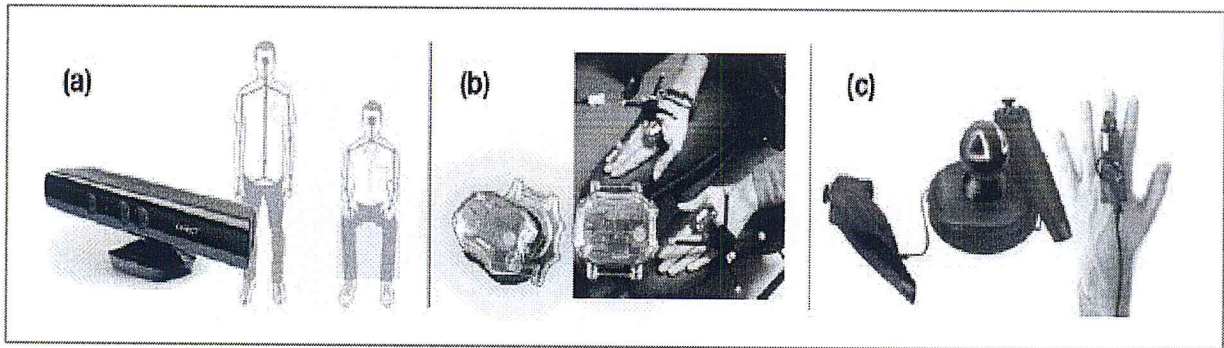


Fig.II.3: Capteurs grand public. (a) Caméra de profondeur Microsoft Kinect, (b) Bracelet à centrale inertielle MoveaMotionPod, (c) Manette à capteur magnétique Razer Hydra et modification possible.

7. 1-2 Représentation des données : traitement et descripteurs gestuels

Le choix des descripteurs gestuels, aussi appelés primitives de représentation du geste, est important, car il influence directement la qualité et la robustesse du système de reconnaissance de gestes. Ces primitives possèdent plusieurs propriétés, comme la continuité, la réversibilité, la sensibilité, l'indépendance et l'homogénéité temporelles. Les auteurs proposent de regrouper ces primitives en quatre catégories [15] :

- Les primitives directes qui sont principalement les données brutes de capteur normalisés.
- Les primitives géométriques comme la distance parcourue, une boîte englobante, la courbure...etc.
- Les primitives visuelles comme les pixels pour un capteur vision.
- Les primitives cinématiques comme la vitesse ; et l'accélération.

Les primitives choisies sont ensuite transmises au système d'apprentissage artificiel sous forme de vecteurs de primitives.

En résumé, l'acquisition du geste peut se faire par différents capteurs. Il existe aujourd'hui un certain nombre de produits grand public offrant des données pertinentes sur le mouvement tout en restant accessibles financièrement.

Cependant, le choix des descripteurs relève de l'activité et des besoins finaux. La caractérisation peut être quantitative, qualitative ou une combinaison des deux.

De plus, ces descripteurs ont une influence sur la qualité de la reconnaissance. Dans la partie suivante, nous détaillons les principales méthodes de reconnaissance du geste.

7. 1-3 Classification :

La tâche de classification consiste selon les cas, soit à déterminer si une action (caractérisée par un descripteur) appartient ou non à une classe de mouvement (détection), soit à déterminer la classe la plus probable d'une action parmi un ensemble de classes (reconnaissance).

La phase de classification ou de reconnaissance permet de transcrire les vecteurs de primitives en symboles. Ces symboles sont associés à des classes de gestes prédéfinis qui sont le vocabulaire de gestes de l'application. Il existe plusieurs méthodes pour modéliser les données sous forme de classes [15].

Fournir au système plusieurs vecteurs de primitives d'un même geste permet d'intégrer les variations éventuelles, évitant d'avoir une classe trop restrictive.

Types d'apprentissage :

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient :

➤ L'apprentissage non supervisé :

L'apprentissage non supervisé, encore appelé apprentissage à partir d'observations ou découverte, consiste à déterminer une classification « sensée » à partir d'un ensemble d'objets ou de situations données (des exemples non étiquetés). On dispose d'une masse de données indifférenciées, et l'on désire savoir si elles possèdent une quelconque structure de groupes. Il s'agit d'identifier une éventuelle tendance des données à être regroupées en classes [16].

➤ L'apprentissage supervisé :

Dans le cas de l'apprentissage supervisé (voir Fig.II.4), on dispose d'un ensemble de données étiquetées, ou d'exemples qui se sont vus associés une classe par un professeur ou un expert. Cet ensemble d'exemples constitue la base d'apprentissage. Les méthodes d'apprentissage supervisé se donnent alors comme objectif général de construire à partir de la base d'apprentissage, des classificateurs, ou fonctions de classement. Une telle fonction permet, à partir de la description d'un objet, de reconnaître un attribut particulier [16].

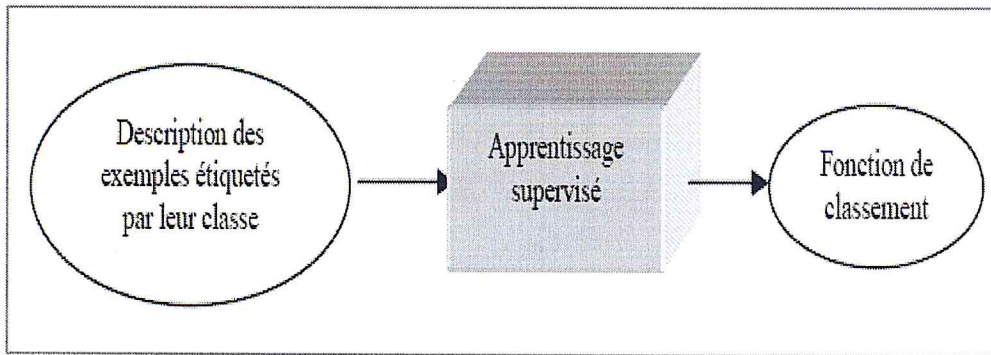


Fig.II.4: L'apprentissage supervisé [16].

Il existe d'autres méthodes d'apprentissage artificiel que l'apprentissage supervisé. [15] distinguent l'apprentissage par renforcement qui consiste à ne pas disposer d'exemples a priori, mais d'acquérir l'exemple durant l'interaction. Si l'utilisateur intervient dans l'apprentissage en fournissant des exemples a priori et pendant l'interaction, ils parlent d'apprentissage actif. Enfin, si les exemples sont fournis pendant l'interaction et sans exemple a priori, ils parlent de façonnage interactif.

D'autres techniques d'apprentissage artificiel existent. Dans le cas d'apprentissage supervisé probabiliste, plutôt que de donner une classe, c'est une probabilité d'appartenance qui est attribuée à chacune des classes prédéterminées. Dans un apprentissage non supervisé, le système effectue l'apprentissage à partir de la masse de données reçues et extrait lui-même les différentes classes selon des critères paramétrés à l'avance. L'apprentissage semi-supervisé combine à la fois l'extraction de classes de façon supervisée et automatique (en utilisant des données non-étiquetées, en combinaison avec des données étiquetées). Le but est d'améliorer la qualité de l'apprentissage. Il existe enfin l'apprentissage par renforcement (qui joue le rôle de collectionneur de problèmes et d'expériences) qui permet de donner plus de poids à certains échantillons, de façon manuelle ou automatique [15].

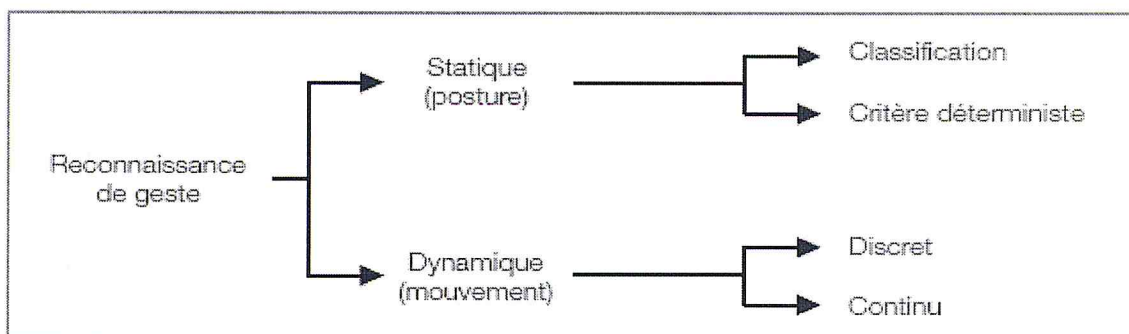


Fig.II.5: Techniques possibles de reconnaissance de gestes [15].

Nous présentons quelques principales méthodes de reconnaissance de geste (voir Fig.II.7). Nous les détaillons dans les prochains paragraphes en distinguant en premier lieu reconnaissance de geste statique et dynamique.

(a) Classification de gestes statiques ou postures :

Pour la détection des gestes statiques ou postures, nous retrouvons essentiellement dans la littérature les classificateurs (ils représentent une famille d'algorithmes de classement statistique. Le rôle d'un classificateur est de classer dans des classes les échantillons qui ont des propriétés similaires, mesurées sur des observations) et la comparaison de prototype (Template matching). Ils permettent d'associer les classes de gestes aux propriétés similaires [15].

(b) Reconnaissance de gestes dynamiques par apprentissage :

Dans le cas d'une reconnaissance de gestes dynamiques, les classes représentent des gestes dynamiques continus. Une fois que le modèle acquies les classes, par exemple : trois classes dans l'illustration (a) de la (voir Fig.II.8), le geste est reproduit et arrive en entrée du système de reconnaissance (b). Celui-ci donne en sortie par comparaison les différentes classes du geste (c).

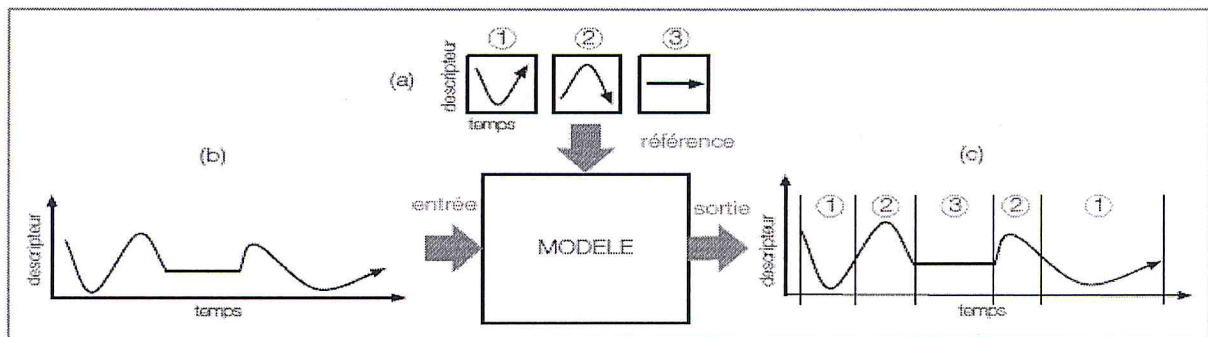


Fig.II.6: Association de classe de gestes a un geste continu, extrait de (a)apprentissage de trois gestes de référence. (b) reproduction d'un geste en continu en entrée du modèle. (c) extraction des classes de gestes par le système de reconnaissance [15].

(c) Machine à Vecteurs de Support :

Les Séparateurs à Vaste Marge ou Support Vector Machines (SVM) ont été proposés en 1995 par V. Vapnik dans son livre « The nature of statistical learning theory » [18]. Ce sont des techniques largement répandues en apprentissage statistique, elles ont eu beaucoup de succès dans quasiment tous les domaines où elles ont été appliquées, et elles ont été fréquemment exploitées dans les domaines de reconnaissance de mouvement humains.

Elles reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. Cette méthode est donc une alternative récente pour la classification [17].

Elle repose sur l'existence d'un classificateur dans un espace approprié.

Étant donné que c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (Kernel) qui permettent une séparation optimale des données.

Le SVM est concerné par l'apprentissage supervisé. Les exemples particuliers sont représentés par un ensemble de couples d'entrées/sorties. Le but est d'apprendre une fonction qui correspond aux exemples et qui prédit les sorties pour les entrées qui n'ont pas encore été examinées. Les entrées peuvent être les descriptions d'objets et les sorties, la classe des objets donnés en entrée.

7.1-4 Interprétation :

À l'issue du processus de classification ou reconnaissance, les éléments symboliques « en sortie » sont interprétés en fonction du contexte. Il peut s'agir d'une probabilité d'appartenance à une classe de geste ou d'une similarité. De plus, certaines caractéristiques du geste peuvent être transmises en cours de son suivi comme la vitesse ou le pourcentage d'avancement par rapport au geste original. Chaque geste reconnu peut être associé à des commandes afin de déclencher des événements ou des contrôles plus fins pour réaliser des PCV (primitives comportementales virtuelles), qui sont :

- L'observation;
- Le déplacement ou la navigation;
- L'action sur le monde virtuel;
- La communication avec autrui ou avec l'application.

La notion de contexte est importante pour plusieurs raisons. Pour l'utilisateur cela permet de réutiliser le vocabulaire gestuel adéquat dans la situation appropriée. Par exemple, dans [15] proposent un moteur de décision qui prend en compte le contexte d'application (comme la zone de l'environnement virtuel où une tâche précise serait à effectuer) et l'historique des commandes ou leurs prédictions.

Le prototype de [15] permet notamment de désigner un objet de la scène par des gestes iconiques. Dans le cas d'une activité, l'interaction gestuelle comporte donc un retour qui

permet au sujet de comprendre son avancement dans l'activité. Il demeure important de lui renvoyer des informations relatives à la compilation de la tâche, par des rétroactions aussi appelées feedbacks.

7.2 Applications des Systèmes de Reconnaissance de Gestes :

Nous allons évoquer quelques exemples d'applications de la reconnaissance de gestes. On dénombre trois catégories d'applications :

- Les applications de surveillance.
- Les applications de contrôle (interfaces pour les jeux vidéo, pour les environnements de réalité virtuelle ou plus généralement interfaces homme-machine).
- Les applications d'analyse (diagnostics médicaux, optimisations des performances d'un athlète, annotation automatique de documents, compression vidéo...).

Trois applications principales sont brièvement présentées dans ce qui suit :

➤ Interaction homme-machine :

A l'heure actuelle, les interfaces de communication les plus utilisées de l'homme vers la machine sont toujours le clavier et la souris. Pour que des systèmes intelligents soient capables d'interpréter efficacement et précisément les gestes complexes de l'homme et permettre une interaction naturelle entre l'homme et la machine (et remplacer peut-être le clavier et la souris), beaucoup de problèmes restent à résoudre.

Les applications qui profiteraient (ou profitent déjà) d'interfaces gestuelles sont, par exemple les applications permettant la visualisation de (grands volumes de) données, la navigation et le contrôle en environnement virtuel et la conception assistée par ordinateur (CAO). Une interface gestuelle en CAO permet de manipuler directement des objets virtuels ou des outils virtuels nécessaires à la réalisation des objets; l'utilisateur se contente ainsi de reproduire les gestes qu'il ferait naturellement pour manipuler réellement les outils et les objets. L'un des avantages est l'apprentissage facilité pour l'utilisation des outils informatiques. Concernant la quête de l'immersion totale en réalité virtuelle, elle passe forcément par une manipulation et une navigation naturelle en environnement virtuel. Quant à la visualisation de données, on commence à voir les limites de l'interaction classique clavier-souris qui permet d'interagir avec un seul objet à la fois. Evidemment, il faut que les systèmes logiciels sous-jacents soient capables de gérer ces nouvelles interactions pour que leurs succès soient garantis. [16]

➤ **Langages des signes :**

L'une des applications évidentes lorsqu'on pense à la reconnaissance de gestes est la reconnaissance du langage des signes. De nombreuses recherches ont été menées autour de l'interprétation de tels langages.

Le langage des signes consiste généralement en trois composants principaux [19] :

- Un alphabet dactylogique (alphabet de langage des signes, finger-spelling).
- Un vocabulaire de signes (word-level sign vocabulary).
- Des caractéristiques non manuelles.

L'alphabet dactylogique est utilisé pour épeler des mots lettre par lettre (généralement des noms propres ou des mots exclus du vocabulaire de signes).

Le vocabulaire de signes est formé de signe représentant des mots et est utilisé majoritairement pour la communication. Les caractéristiques non manuelles consistent en expressions faciales, à la position de la langue, de la bouche et du corps.

L'intérêt du langage des signes en reconnaissance de gestes est sa structure relativement précise, permettant la définition de règles contextuelles et grammaticales strictes pouvant être appliquées pour faciliter la reconnaissance [16].

➤ **Systèmes de surveillance :**

Dans [20] les auteurs présentent le besoin de systèmes de surveillance comme la principale raison de l'avancée dans le domaine de la capture et la reconnaissance de mouvements corporels. Ces systèmes ont pour but d'analyser les comportements humains et de détecter de manière automatique des comportements hors normes comme le fait de commettre un crime, par exemple un vol de voiture. Les systèmes de surveillance automatiques doivent répondre à plusieurs défis importants de la reconnaissance de gestes ; initialisation automatique, reconnaissance de lieu, segmentation de sujets dans une foule... Des applications dans ce domaine pourraient par exemple compter les individus d'une foule, analyser la congestion ou les comportements dans une file d'attente ou identifier des individus.

8. Conclusion :

Les applications à interfaces gestuelles peuvent donc s'insérer dans des applications courantes, en faisant appel à une sélection d'outils vus dans ce chapitre. La disponibilité des algorithmes efficaces et robustes est un fait actuel, mais réaliser un ensemble de gestes concernés par l'identification n'est pas encore facile. Donc pour avoir des résultats intéressants, des gestes iconiques doivent être choisis.

Dans le chapitre suivant nous allons aborder le but du sujet qui est la conception où nous allons détailler toutes les étapes de la conception de notre système de reconnaissance de geste.

Chapitre III: Conception et Réalisation

1. INTRODUCTION

Dans ce chapitre nous allons aborder le problème de la reconnaissance des gestes d'une personne à partir des images acquises. Il s'agit d'un problème extrêmement difficile, en raison de la complexité de l'objet (un être humain) qui est observé. Le corps humain est un objet hautement articulé : un modèle cinématique du corps, même simplifié, contient au minimum une vingtaine de degrés de liberté.

Les configurations possibles du corps sont donc extrêmement nombreuses. La variabilité d'apparence entre différentes personnes est aussi très forte. La même posture prise par différents individus génère des observations très différentes en raison des différences de taille, de l'aspect visuel. A cela, s'ajoutent les ambiguïtés visuelles introduites par le mode d'observation. Nous cherchons à estimer la position des membres dans l'espace à partir de la projection du capteur sur le corps, et même si nous disposons de plusieurs points de vue, une partie de l'information sur la profondeur est nécessairement perdue. L'apparence du corps dans une image varie aussi en fonction du point de vue ou de l'éclairage.

Le processus de reconnaissance de formes, est proche du raisonnement de l'être humain, d'où s'inspirent les domaines de l'intelligence artificielle, qui servent à automatiser des tâches humaines.

Donc durant ce chapitre nous abordons la démarche adoptée pour la réalisation de notre système de reconnaissance de gestes, ou nous allons présenter la base de données utilisée pour le stockage des informations ainsi que toutes les étapes franchies pour la conception de notre système.

2. Démarche adoptée pour concevoir le système de reconnaissance de gestes :

Cette thèse présente un système qui fait une analyse robuste en temps réel qui se base sur un ensemble de règles. Ces règles sont appliquées sur les différentes jointures du squelette du corps humain, qui est généré par le capteur Kinect, en temps réel. A travers ces règles le système définit la situation actuelle des articulations du corps (gestes dynamiques et postures) pour l'interprétation de scène et la vidéo surveillance.

Pour concevoir ce système de reconnaissance gestuelle (Fig.III.1), il faut spécifier d'abord un vocabulaire gestuel, Ce dernier dépendra de la nature des gestes d'interaction. Nous avons opté pour des gestes statiques (la pose) et dynamiques, c'est-à-dire que nous prenons en considération le positionnement du corps humain au fil du temps.

Pour la conception de notre système, nous avons suivi les étapes classiques (d'après la littérature) d'un système de reconnaissance de formes, qui sont présentées par le schéma suivant:

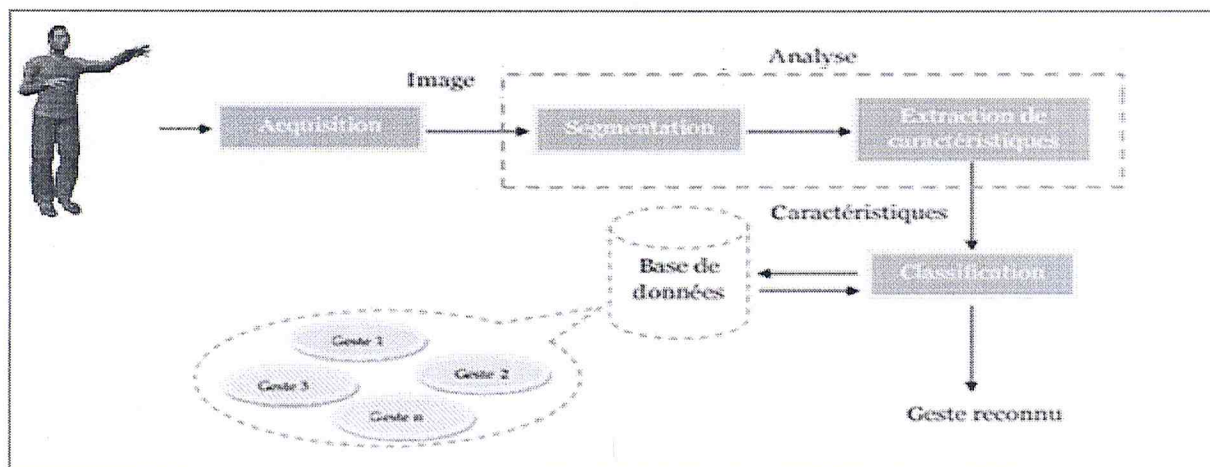


Fig.III.1: Schéma d'un système de reconnaissance de geste.

2.1 Création de notre base de données des gestes:

Voici à présent notre base de gestes avec laquelle nous avons fait l'apprentissage et à partir de laquelle, chaque geste acquis sera comparé afin de le reconnaître, pour cela nous avons besoin d'un fichier XML qui décrit les gestes que nous voulons évaluer.

➤ Définition d'XML :

Le langage XML est un langage qui permet de décrire des données à l'aide de balises et de règles que l'on peut personnaliser, on peut dire aussi qu'il est un langage informatique de balisage générique [21].

Un langage de balisage est un langage qui s'écrit grâce à des balises. Ces balises permettent de structurer de manière hiérarchisée et organisée les données d'un document. Le terme générique signifie que nous allons pouvoir créer nos propres balises. Nous ne sommes pas obligés d'utiliser un ensemble de balises existantes :

- **Les balises**

Les balises sont les éléments de base d'un document XML. Une balise porte un nom qui est entouré de chevrons. Une balise commence donc par un < et se termine par un >[21].

En XML, on distingue deux types de balises : les balises par paires et les balises uniques.

a. Les balises par paires

Les balises par paires sont composées en réalité de deux balises que l'on appelle ouvrantes et fermantes, la balise ouvrante commence par < et se termine par > tandis que la balise fermante commence par </ et se termine par >[21].

Une balise par paires ne peut pas contenir n'importe quoi : elle peut contenir une valeur simple comme par exemple une chaîne de caractères, un nombre entier ou un nombre décimal[21].

Enfin, une balise par paires peut contenir un mélange de valeurs simples et de balises comme en témoigne l'exemple suivant :

```
<Gestes>
  <Geste Description="ZoomAvant" TempsExecutionMax="1000" CodeCle="ADD" Message="" Chemin="">
    <ComposantGeste DebutRelation="ADroiteDe" DistanceD="0,2" FinRelation="AGaucheDe"
      DistanceF="-0,2" PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" />
    <ComposantGeste DebutRelation="AGaucheDe" DistanceD="-0,2" FinRelation="ADroiteDe"
      DistanceF="0,2" PremiereJoint="WristLeft" DeuxiemeJoint="ShoulderLeft" />
    ...
  </Geste>
  ...
</Gestes>
```

Fig.III.2:Exemple de document XML.

b. Les balises uniques

Une balise unique est en réalité une balise par paires qui n'a pas de contenu[21].

• Les attributs

Il est possible d'ajouter à nos balises ce qu'on appelle des attributs. Un attribut peut se décrire comme une option ou une donnée cachée. Ce n'est pas l'information principale que souhaite transmettre la balise, mais il donne des renseignements supplémentaires sur son contenu.

- La valeur d'un attribut doit impérativement être délimitée par des guillemets, simples ou doubles[21].
- Dans une balise, un attribut ne peut être présent qu'une seule fois[21].

➤ XML en tant que base de données

Un document XML (Fig.III.2) peut être considéré comme une base de données uniquement au sens le plus strict de l'expression, à savoir, une collection de données. XML en cela n'est guère différent des autres fichiers, car après tout, tous les fichiers contiennent d'une certaine manière des données. En tant que format de "base de données", XML présente cependant certains avantages. Il est par exemple auto-descriptif car les balises décrivent la structure et le type des noms des données

La figure (Fig.III.3) présente quelques exemples de notre base de geste.

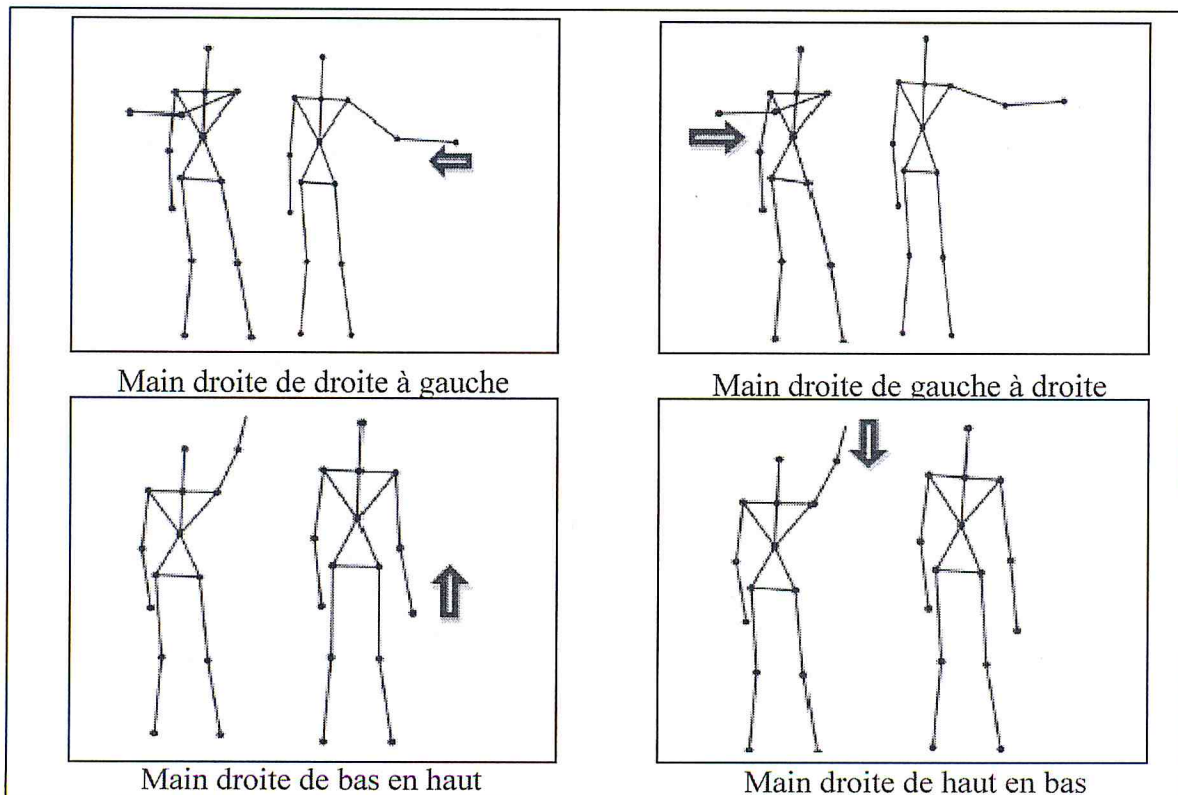


Fig.III.3: Exemples de gestes pris en compte par notre base de données.

2.2 Les étapes classiques d'un système de reconnaissance de formes :

Les étapes que nous avons suivies pour la conception de notre travail sont détaillées dans ce qui suit :

2.2.1 Etape 1 : Acquisition du geste :

Dans le chapitre précédent, nous avons cité les différents outils d'acquisition des gestes du corps humain, et notre choix s'est porté sur le capteur Kinect, parce que c'est l'un des meilleurs moyens d'acquisition de la silhouette de l'être humain et de manière rapide.

Le choix de ce dispositif est dû au fait, que nous nous intéressons à la forme du corps humain avec tous ses détails ainsi la Kinect nous fournit les données nécessaires en nous fournissant les données des coordonnées des différents jointures du corps.

La Kinect, développée par Microsoft, est un périphérique utilisé largement dans les jeux et par les communautés de recherche. Elle peut capturer les images RGB et de profondeur. Avec l'information de la profondeur, il n'est plus difficile de détecter et de traquer le corps de l'utilisateur, dans un environnement bruyant ou encombré. A cause de cette performance attirante et un prix raisonnable, elle peut être largement utilisée dans diverses tâches de vision comme le suivi du visage, le suivi des mains, la reconnaissance des actions de l'humain, et la reconnaissance des gestes.

a) La Kinect :

La Kinect est un périphérique basé sur une technologie logicielle développée par Rare, une filiale de Microsoft Game Studios appartenant à *Microsoft*, et sur une caméra spécifique créée par PrimeSense, qui interprète les informations sur la scène 3D obtenue à travers une lumière laser infrarouge structurée et projetée en continu. Ce système de scanner 3D appelé « Light Coding » utilise une variante de la reconstruction 3D basée sur l'image [19].

La Kinect (Fig.III.4) a été créé initialement pour le domaine des jeux vidéo, elle permet à l'utilisateur de contrôler le jeu avec les mouvements de son corps ou exécuter des commandes à travers la voix (des paroles prédéfinies) sans utiliser de manette [19].

Le capteur Kinect est une barre horizontale reliée à une petite base avec un pivot motorisé, conçu pour être placé au-dessus ou en dessous de l'affichage vidéo (téléviseur, écran d'un ordinateur).

Le dispositif comporte une caméra RGB, un capteur de profondeur et un microphone multi-réseau exécutant un logiciel propriétaire, qui fournit la capture du mouvement du corps en 3D, la reconnaissance faciale et la reconnaissance vocale [19].

Le capteur de profondeur se compose d'un projecteur laser infrarouge combiné à un capteur CMOS monochrome, qui capture des données vidéo en 3D dans toutes les conditions de lumière ambiante. Le rayon de détection de ce capteur est réglable, et le logiciel de Kinect est capable de calibrer automatiquement le capteur en fonction de la jouabilité ou de l'environnement physique du joueur, pouvant accueillir la présence des meubles ou d'autres obstacles [19].

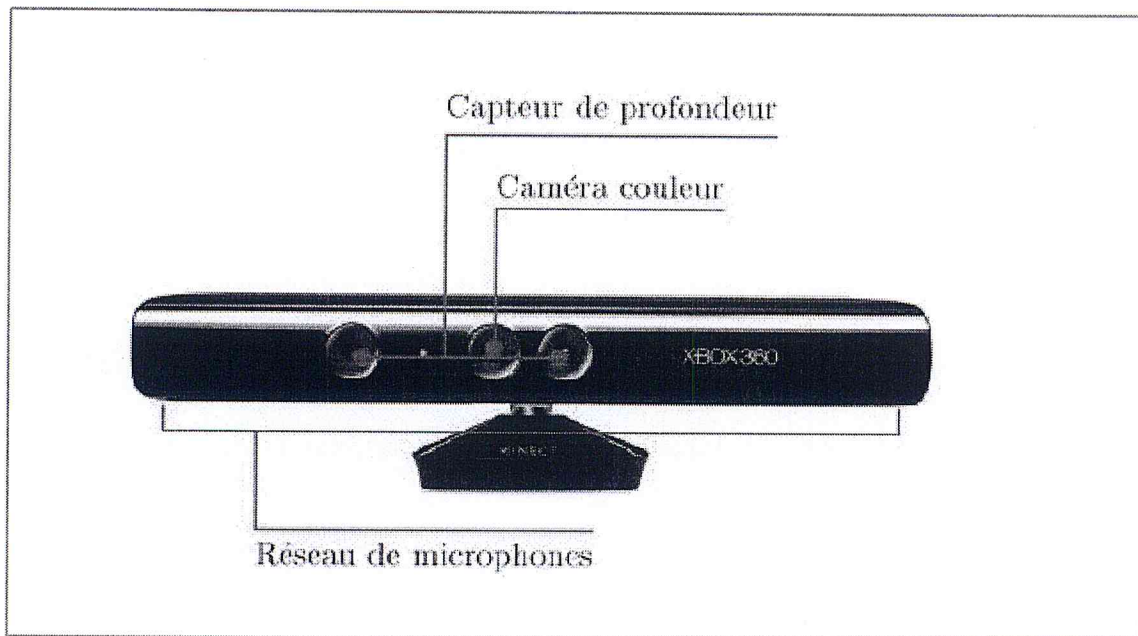


Fig.III.4: Laface avant de la Kinect.

Devant l'étendue des possibilités offertes par la Kinect (Fig.III.4), des logiciels de développement « Open Source » ont été créés pour manipuler la Kinect par ordinateur et créer des applications en utilisant la Kinect.

b) Kinect en chiffre :

La Kinect est sortie le 4 novembre 2010 en Amérique du Nord et le 10 novembre en Europe. Cette nouvelle technologie ne coûtait que 150€/\$.

Plus de huit millions d'exemplaires de la Kinect ont été vendus partout dans le monde deux mois après sa sortie, dont un million en seulement dix jours. Le 11 mars 2011, la Kinect entre au livre Guinness des records comme étant « l'accessoire high-tech le plus vendu dans un court laps de temps » avec 10 millions d'unités vendues, soit une moyenne officielle de 133 333 Kinects vendues chaque jour à travers le monde[22].

c) Caractéristiques de la Kinect :

Voici quelque caractéristique de la Kinect [19] :

Capteurs :

- Caméra couleur
- Capteur de profondeur
- Micro à reconnaissance vocale

Motorisation :

- Ensemble motorisé pour suivre les déplacements
- Inclinaison: ± 27 degrés

Champ de vision :

- Champ de vision horizontal : 57 degrés
- Champ de vision vertical : 43 degrés
- Portée du capteur : 0,8 m – 4,0 m

Flux de données :

- Infrarouge 640x480 à 30 images par seconde
- RVB 640x480 à 30 images par seconde
- Audio 16 bits à 16 kHz

Système de reconnaissance physique :

- Jusqu'à 6 personnes et 2 joueurs actifs
- 20 articulations par squelette

Audio :

- Chat vocal Xbox Live et chat vocal dans les jeux (nécessite un compte Xbox Live Gold) ;
- Suppression de l'écho ;
- Reconnaissance vocale multilingue ;

d) Les différents composants de la Kinect :

Comme le montre (Fig.III.5), la Kinect est composée de plusieurs capteurs audio, d'une caméra RVB (Rouge, Vert, Bleu), d'un projecteur infrarouge, d'une caméra infrarouge, et d'un pied motorisé.



Fig.III.5: Les composants de la Kinect [22].

Grâce à tous ces composants la Kinect fournit trois flux différents qui sont le flux audio, le flux vidéo, et le flux de profondeur.

✓ Le flux audio :

Le flux audio provient des capteurs audio et permet la fonctionnalité de reconnaissance vocale. Dans notre travail, ce flux n'est pas utilisé [22].

✓ Le flux vidéo :

Le flux vidéo provient de la caméra RVB qui fonctionne comme toutes les autres caméras 2D. Elle fournit des images de résolution 640x480 à un taux de 30 images par seconde. Ce flux est très important car il permet de fournir des images couleurs captées par la Kinect [22].

✓ Le flux de profondeur :

Le flux de profondeur est le plus important car il fait l'originalité de la Kinect en la différenciant des habituelles caméras vidéo. Ce flux fournit une carte de profondeur donnant pour chaque pixel la distance dans l'espace 3D entre l'objet représenté par ce pixel et la caméra. On peut facilement transformer la carte de profondeur en image de profondeur en niveau de gris pour visualiser la profondeur [22].

La carte de profondeur est calculée grâce au projecteur infrarouge et à la caméra infrarouge en utilisant le principe de la lumière structurée (reconstruction 3D) [22].

e) Intérêt de la Kinect par rapport à une caméra classique

L'image de profondeur est plus facile à analyser pour une machine qu'une image en couleur classique.

Image couleur classique[23]:

1. La reconnaissance d'objets et de personne passe par l'analyse des pixels...
2. Valeur des pixels dépend des conditions de prise de vue : éclairage, ouverture, modification des couleurs par l'appareil, etc...
3. Impossibilité de connaître facilement le contenu d'une image, de savoir où commence et où finit un objet, etc...

Image en profondeur[23]:

1. L'intensité des pixels est proportionnelle à la distance de l'objet considéré avec la caméra.
2. L'intensité des pixels est indépendante de l'éclairage
3. Possibilité de localiser avec précision les objets dans la scène, ou ils commencent / finissent, etc...
4. Reconstruction 3D de la scène
5. Image de profondeur permet d'appliquer des algorithmes de tracking plus efficace que sur une image classique:
 - Tracking des personnes
 - Tracking des articulations du corps

2.2.2 Etape 2 : Analyse (segmentation et extraction des caractéristiques)

Lorsque nous cherchons à identifier les mouvements humains, il est parfois judicieux de savoir où se trouve l'acteur de l'action afin de l'isoler du reste de la scène. Ceci permet de se focaliser sur ses mouvements, indépendamment de ce qui peut se passer à l'arrière-plan. Pour ce faire, nous avons suivi l'analyse de la forme du personnage, par le biais de sa silhouette.

a) Analyse de la silhouette

Dans les analyses de la forme par des silhouettes, la modélisation de l'acteur peut être : la séparation entre le premier plan et l'arrière-plan permet de considérer le premier plan comme un acteur. Une modélisation plus complexe peut être mise en œuvre : l'identification du personnage est faite par un algorithme de détection de l'humain, la méthode proposée est basée sur le paradigme des exemplars et une extraction des silhouettes, afin de reconnaître des mouvements. Un espace des exemplars est défini par un ensemble d'exemples 'représentatifs', issus du jeu de données d'entraînement, muni d'une fonction de distance entre deux points de cet espace [24].

b) Explication de notre méthode d'analyse de gestes

A partir des données extraites de la Kinect, nous effectuons la segmentation du signal continu pour en extraire les parties correspondant à des gestes et à des postures. La difficulté ici réside dans le fait qu'il est difficile de trouver un critère objectif pour déterminer le début et la fin de mouvement au sein de la séquence vidéo candidate.

Au cours des différentes recherches, nous avons opté pour la solution définie dans ce qui suit :

En formant un squelette qui représente l'individu détecté à travers Kinect, nous pouvons avoir les coordonnées (x, y, z) des différentes jointures du corps (qui sont 20 au total (Fig.III.6) dont nous allons utiliser seulement 15 d'entre elles). Nous avons suivi des méthodes alternatives à celle utilisées dans la version originale pour l'activation des flux des données de la Kinect, car nous utilisons le Windows Application Forms (WinForms), qui, contrairement à Windows Presentation Foundation (WPF), n'a pas les mêmes outils surtout ceux qui sont en relation avec Kinect (Ces deux dernières notions vont être définies dans le chapitre 4).

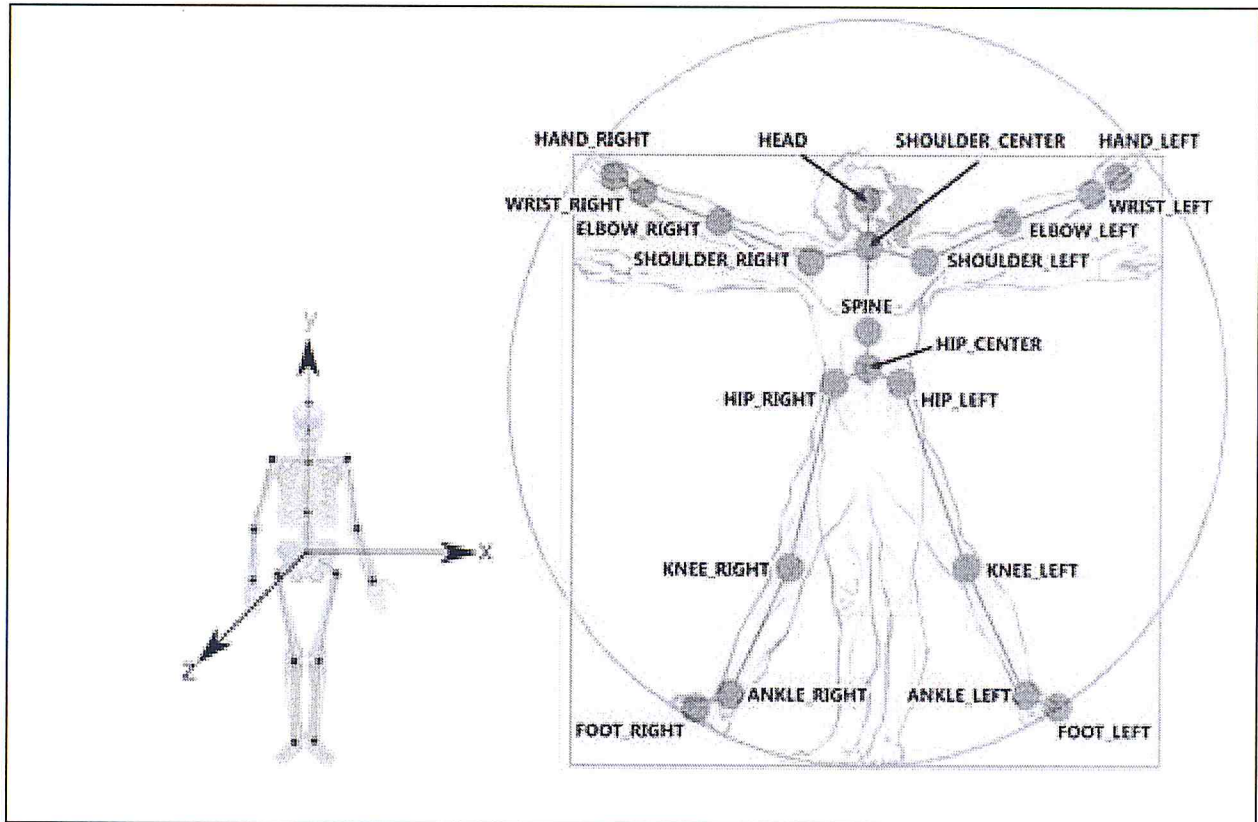


Fig.III.6: Positions des articulations du corps humain détectés.

Notre approche se base sur le test de ces coordonnées à travers un ensemble de règles pour définir les gestes dynamiques et statiques actuels du squelette. Une réception en continue des coordonnées est requise pour l'évaluation de la situation du squelette en temps réel. L'ensemble de règles, qui permettent de définir l'état du corps, peuvent être catégorisé en trois :

➤ **Règles pour les gestes dynamiques :**

On a défini le geste dynamique comme la succession de deux postures à deux moments différents et qui vérifient une certaine règle. Ces règles vérifient l'état d'une jointure du squelette par rapport à une autre jointure à t_0 et t_1 (tel que le temps t_0 est différent de t_1) sans dépasser un temps maximal pour l'exécution de ce geste (Fig.III.7). Si la règle est vérifiée alors soit elle affiche un message, lance un exécutable depuis un chemin défini, simule une touche du clavier ou une combinaison de ces trois options.

```

<Gestes>
  <Geste Description="Glisser main droite vers la gauche" TempsExecutionMax="5000"
    CodeCle="ADD" Message="" Chemin="">
    <ComposantGeste DebutRelation="ADroiteDe" DistanceD="0,15"
      FinRelation="AGaucheDe" DistanceF="0"
      PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" />
    ...
  </Geste>
  ...
</Gestes>

```

Fig.III.7:L'exemple du geste « Glisser main droite vers la gauche ».

Voici l'exemple du « Glisser main droite vers la gauche » (Fig.III.7) qui a une première

Jointure« WristRight » (Poignet droit) et une deuxième jointure« ShoulderRight » (Epaule droite). Ce geste est vérifié si l'individu fait un geste avec sa main droite en la déplaçant de la droite de l'épaule droite vers sa gauche,(Elle commence avec la relation début 'ADroiteDe' et finit par la relation 'AGaucheDe',les différentes relations sont listées dans la Fig.III.8)accompagné, d'une distance, donc dans ce cas :

```

public enum RelationJoint
{
    Aucun,
    AuDessus, //Concerne la coordonnée X
    AuDessous, //Concerne Y
    ADroiteDe, //Concerne X
    AGaucheDe, //Concerne X
    DroiteHaut, //Concerne X et Y
    DroiteBas, //Concerne X et Y
    GaucheHaut, //Concerne X et Y
    GaucheBas, //Concerne X et Y
    VersAvant, //Concerne Z
    VersArriere //Concerne Z
}

```

Fig.III.8:La classe des différentes relations

PositionXmain>PositionX épaule + 0,15) vers sa gauche (PositionXmain<PositionX épaule) sous un laps de temps de 5 secondes (5000 Millisecondes).

Si ce geste est vérifié donc il simulera seulement le bouton '+' du clavier venant de l'attribut « CodeCle », puisque les deux autres attributs « Message » et « Chemin » sont vides. (L'attribut « Message » est pour l'affichage d'un message et « Chemin » pour montrer l'adresse d'un exécutable sur l'ordinateur).

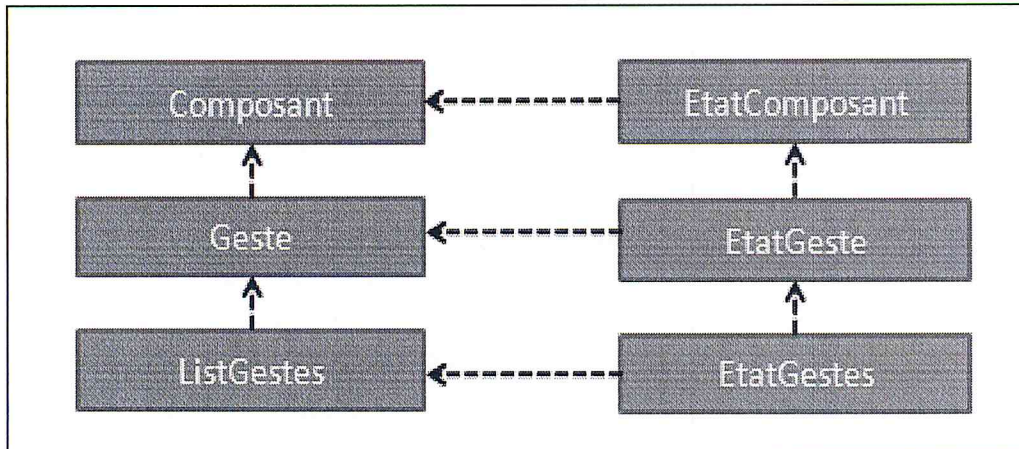


Fig.III.9:Schéma du programme qui gère les gestes.

La figure précédente (Fig.III.9) montre le schéma bloc du programme qui gère les gestes dynamiques :

Composant : Il peut être considéré comme une condition pour un certain geste (C'est la balise « ComposantGeste » dans la (Fig.III.7) Il décrit le début et la fin d'une relation entre deux jointures.

EtatComposant : C'est une classe qui vérifie l'état du composant, dans ce cas : elle vérifie si le début et la fin de la relation sont satisfaits ou non.

Geste : Il regroupe un ensemble de composant qui forme un geste quelconque, il contient sa description, son temps d'exécution maximale avec sa commande (son but). Il est représenté par son même nom dans la (Fig.III.7) avec la balise « Geste ».

EtatGeste : Comme la classe EtatComposant mais ici il vérifie s'il mémorise le temps du début d'exécution et vérifie s'il a dépassé ou non son temps d'exécution demandé en vérifiant l'état des composants (Si le début et la fin des composants sont satisfaits ou non).

ListGestes : Elle regroupe tous les gestes et s'occupe du chargement du fichier XML des gestes dynamiques.

EtatGestes : Elle contient la fonction appelée du programme principal, qui dépend de la réponse de la classe EtatGeste, et qui exécute l'action du geste correspondant ou non.

➤ **Règles pour les gestes statiques (ou postures/pose) :**

Nous avons défini les gestes statiques (ou postures) de la même façon que les gestes dynamiques, sauf que dans ce cas c'est les positions des jointures à un instant fixe. Les règles des postures vérifient l'état d'une jointure du squelette par rapport à l'autre à un temps quelconque. La règle de la posture peut être exécutée directement ou retardée avec un temps donnée (par exemple :

mettre la main droite en haut pendant 15 secondes) (Fig.III.10). Si la règle est vérifiée alors il affiche le nom de la posture.

```

<Postures>
  <Posture Description="Assis" Visible="True" TempsAttenteExecution="0">
    <Condition PremiereJoint="KneeLeft" DeuxiemeJoint="HipCenter"
      Relation="VersAvant" Distance="-0,2" />
    <Condition PremiereJoint="KneeRight" DeuxiemeJoint="HipCenter"
      Relation="VersAvant" Distance="-0,2" />
  </Posture>

  <Posture Description="Haut la main droite!" Visible="True" TempsAttenteExecution="0">
    <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter"
      Relation="AuDessus" Distance="0" />
  </Posture>

  <Posture Description="Haut la main gauche!" Visible="True" TempsAttenteExecution="0">
    <Condition PremiereJoint="WristLeft" DeuxiemeJoint="HipCenter"
      Relation="AuDessus" Distance="0" />
  </Posture>

  <Posture Description="Main droite vers l'avant" Visible="True" TempsAttenteExecution="5000">
    <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter"
      Relation="VersAvant" Distance="0" />
  </Posture>
</Postures>

```

Fig.III.10 : Exemples sur des postures.

Voici quelques exemples sur des postures. Prenons l'exemple de la posture « Assis » qui possède deux conditions, la première elle vérifie si le genou gauche se trouve vers l'avant par rapport au bassin avec une distance de -0.2mètre, la deuxième condition vérifie la même chose pour le côté droit. Si ces deux conditions sont vérifiées donc l'individu est assis.

Un autre exemple, la posture « Main droite vers l'avant » possède une condition simple sauf qu'elle est différente des autres avec son retard d'exécution (L'attribut 'TempsAttenteExecution'), ici, si l'individu laisse sa main droite vers l'avant pendant 5 secondes, la posture sera vérifiée.

En plus, des attributs précédents nécessaires, nous avons ajouté un attribut « Visible » qui contrôle si cette posture s'affiche dans le champ d'affichage des postures, tel que l'ajout d'une posture pour le but unique d'être utilisé dans les scénarios que nous allons voir après cette partie.

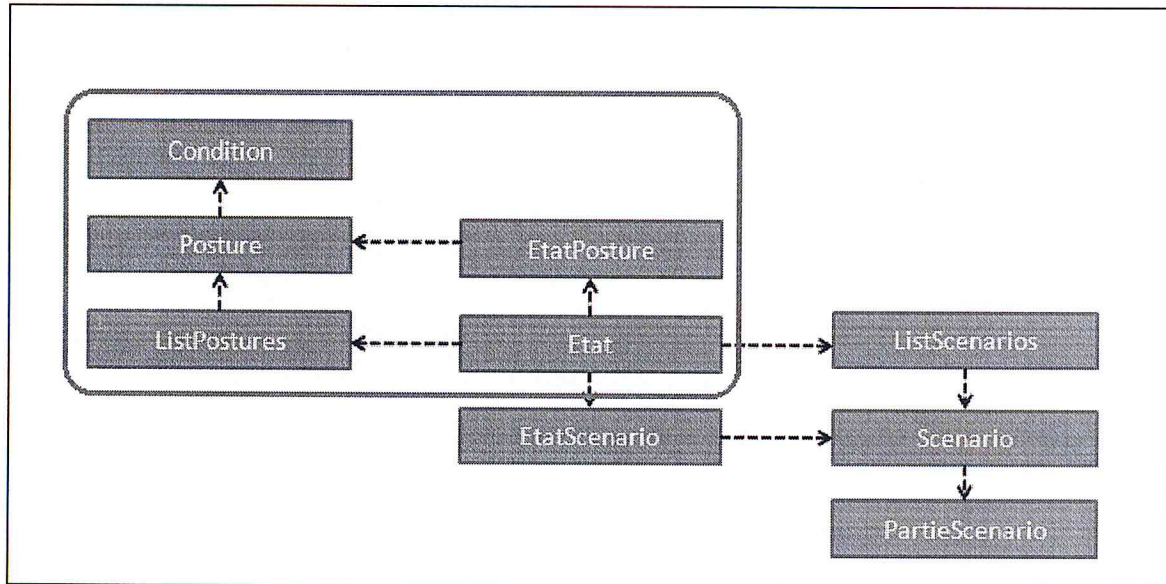


Fig.III.11:Schéma du programme qui gère les postures et les scénarios.

La figure (Fig.III.11) montre le schéma bloc du programme qui gère les postures et les scénarios, mais nous allons se concentrer ici sur la partie encadrée en rouge (partie posture) :

Condition : Comme son nom l'indique c'est une condition qui est testée pour la vérification d'une certaine posture (La balise correspondante c'est « Condition » dans la figure (Fig.III.11)). Il décrit la relation entre deux jointures à n'importe quel moment.

Posture : Elle regroupe un ensemble de conditions, elle contient sa description et un temps de retard pour l'exécution si c'est nécessaire. Elle est représentée par son même nom dans la figure (Fig.III.11) avec la balise « Posture ».

EtatPosture : Dans le cas où une posture a un retard avant son exécution, cette classe gère le temps d'attente, et exécute la posture après la fin d'attente.

ListPostures : Elle regroupe toutes les postures et s'occupe du chargement du fichier XML des postures.

Etat : Elle contient la fonction appelée du programme principal, et c'est ici où elle gère la vérification des conditions pour chaque postures.

➤ Règles pour les scénarios :

Quant aux scénarios, nous les avons définis comme une suite de postures soit en suivant un ordre précis, soit de façon aléatoire. Il s'exécute après avoir satisfait toutes ou un certain pourcentage de postures requises en un temps défini (Voir Fig.III.12).


```

<Scenarios Compteur="10" Seuil="50">
  <Scenario Description="Haut les mains!" Attention="Soupçon" Pourcentage="50"
    Attention2="Rien à voir" TempsExecutionMax="5000" Suspect="True">
    <Partie Nom="Haut la main droite!" Ordre="1" />
    <Partie Nom="Haut la main gauche!" Ordre="2" />
  </Scenario>

  <Scenario Description="Cercle" Attention="Cercle?" Pourcentage="50"
    Attention2="" TempsExecutionMax="5000" Suspect="False">
    <Partie Nom="BrasDroitVersDroitHaut" Ordre="1" />
    <Partie Nom="BrasDroitVersDroitBas" Ordre="2" />
    <Partie Nom="BrasDroitVersGaucheBas" Ordre="3" />
    <Partie Nom="BrasDroitVersGaucheHaut" Ordre="4" />
  </Scenario>

  <Scenario Description="Mouvement" Attention="Attention Mouvement" Pourcentage="50"
    Attention2="Stable" TempsExecutionMax="5000" Suspect="False">
    <Partie Nom="PiedDroitVersAvant" Ordre="0" />
    <Partie Nom="PiedGaucheVersAvant" Ordre="0" />
  </Scenario>
</Scenarios>

```

Fig.III.12: Exemples pour les scénarios

Voici quelques autres exemples pour les scénarios (voire Fig.III.12). Pour l'exemple « Haut les mains ! » qui a deux postures, si l'individu lève sa main droite en premier puis lève sa main gauche avant le passage de 5 secondes, le scénario affichera le message « Haut les mains ! => date actuelle », sinon, si l'individu n'a pas terminé ce scénario complètement et si sa progression dans ce scénario été supérieure au pourcentage défini dans l'attribut « Pourcentage » le scénario affichera le message « Soupçon => date actuelle », sinon, s'il est inférieur affichera « Rien à voir => date actuelle ». La même chose pour les autres exemples sauf dans le cas de l'exemple « Mouvement » qui ne nécessite pas un ordre précis. Chaque scénario peut représenter un cas normal ou anormal, c'est pourquoi nous avons ajouté l'attribut « Suspect » qui définit son cas.

Les attributs restants qui se trouvent dans la tête du fichier XML du « Scenarios » sont en relation avec un algorithme de mesure de similarité que nous allons voir dans la partie « Classification ».

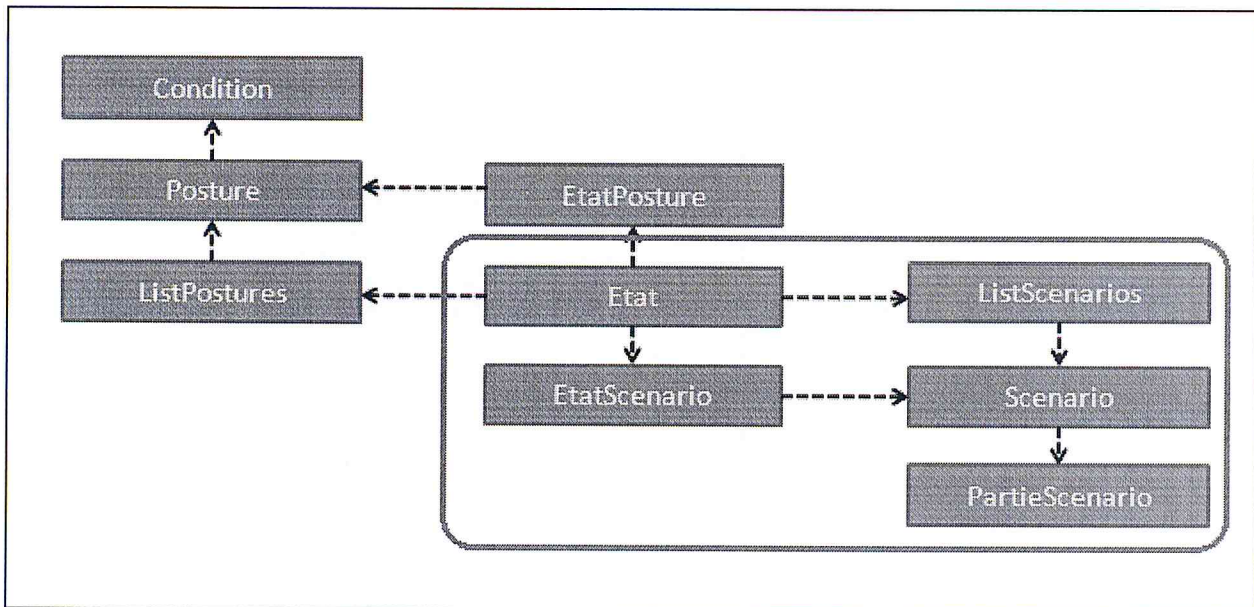


Fig.III.13:Schéma du programme qui gère les postures et les scénarios

La même figure que (Fig.III.11) la figure (Fig.III.13) montre le schéma bloc du programme qui gère les postures et les scénarios, mais maintenant nous allons voir la partie des scénarios :

PartieScenario : C'est une classe qui contribue à la formalisation d'un scénario. Une partie représente une seule posture à la fois qui appartient à un scénario. Elle a le nom de la posture avec une valeur booléenne vraie si elle est vérifiée sinon fausse, et elle a son ordre dans ce scénario (La balise correspondante à ça c'est « Partie » de la figure (Fig.III.12)).

Scenario : Cette classe regroupe un ensemble de postures, elle contient sa description si toutes les postures (parties) sont vérifiées sinon elle affiche le message mis dans l'attribut 'Attention' si le pourcentage est supérieur à un seuil prédéfini. Elle a aussi un temps maximal pour son exécution (Elle représente la balise « Scenario »).

EtatScenario : La même chose comme les postures, sauf qu'ici elle vérifie si le scénario a dépassé son temps maximal d'exécution.

ListScenarios : Elle se charge de regrouper tous les scénarios ainsi que le chargement du fichier XML des scénarios.

Etat : En plus des postures, elle se charge de la vérification des scénarios.

c) Etape 3 : Classification

En plus des règles que nous avons définies pour la distinction des gestes (qu'ils soient dynamiques, des postures ou des scénarios) et leurs interprétations selon la base de données des gestes définies, nous cherchons à calculer la similarité des postures exécutés en temps réel par rapport au scénario et puisqu'on peut considérer ces gestes comme un ensemble de chaînes de caractères (formé à partir des noms des postures) donc nous avons introduit une méthode qui évalue la similarité entre des textes. Nos scénarios sont formés à partir des postures et avec un enregistrement des échantillons de postures en temps réel sous format d'une liste de chaîne de caractères, une méthode de similarité entre deux textes (scénario et posture en temps réel) est donc valable pour notre application. L'enregistrement des postures en temps réel se fait à travers une liste avec un compteur, qui, s'il arrive à un certain nombre il déclenchera la phase du traitement de similarité avec chaque scénario disponible.

➤ Les méthodes de mesure de similarité :

Dans cette contribution, nous avons présenté des mesures de similarité que nous avons jugée apte pour la classification des données textuelles pour notre application.

1) Mesure de similarité « cosinus » :

Évaluer des similarités entre entités textuelles est un des problèmes centraux dans plusieurs disciplines comme l'analyse de données textuelles, la recherche documentaire ou l'extraction de connaissances à partir de données textuelles (Text Mining). Dans chacun de ces domaines, les similarités sont en effet utilisées pour une large variété de traitements[25].

Si on peut relier la similarité entre deux vecteurs A et B (Dans notre cas, la séquence A représente les postures enregistrées en temps réel et la séquence B représente un scénariosauvegardé dans la base de données) à la mesure de l'angle θ qu'ils forment, alors nous pouvons les évaluer en calculant le cosinus de cet angle : c'est ainsi qu'est définie la *similarité cosinus*. Le calcul du cosinus se base sur l'expression du produit scalaire $A \cdot B = \|A\| \|B\| \cos(\theta)$ et implique qu'aucun des deux vecteurs ne soit nul[26].

Pour calculer la similarité du cosinus on doit calculer le produit scalaire des deux vecteurs et le diviser par le produit des normes des deux vecteurs [26], voici la formule :

$$\text{similarité}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Si nos poids ne peuvent pas être négatifs, alors on a $0 \leq \text{similarité}(A, B) \leq 1$. Plus la mesure de l'angle est faible, plus son cosinus est élevé : avec cette formule, plus similarité(A, B) s'approche de 1, plus la similarité entre les objets représentés par les vecteurs A et B est grande [26].

Exemple de calcul

Soit la séquence A = WXZ et la séquence B = WYWZ :

$$\frac{1 \cdot 2 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1}{\sqrt{1^2 + 1^2 + 0^2 + 1^2} \sqrt{2^2 + 0^2 + 1^2 + 1^2}} \simeq 0.72$$

	W	X	Y	Z
A	1	1	0	1
B	2	0	1	1

Tab.III.1 : tableau de répétitions des caractères dans les deux séquences

2) Mesure de similarité (Alignement global de Needleman & Wunsch (1970))

L'algorithme de *Needleman-Wunsch* est un algorithme qui effectue un alignement global maximal de deux chaînes de caractères [27]. Il permet d'aligner une séquence sur la totalité de la longueur de l'autre séquence [27]. Il est couramment utilisé en bio-informatique pour aligner des séquences de protéines ou de nucléotides. L'algorithme a été présenté en 1970 par Saul Needleman et Christian Wunsch [27].

L'algorithme de Needleman-Wunsch est un exemple de programmation dynamique (qui consiste à résoudre un problème en le décomposant en sous-problèmes, puis à résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires). Il garantit de

trouver l'alignement de score maximal. Ce fut la première application de la programmation dynamique pour la comparaison de séquences biologiques[27].

- **Construction de la matrice**

Soit deux séquences *A* et *B* deux séquences de longueurs *m* et *n* (Voire Fig.III.14):

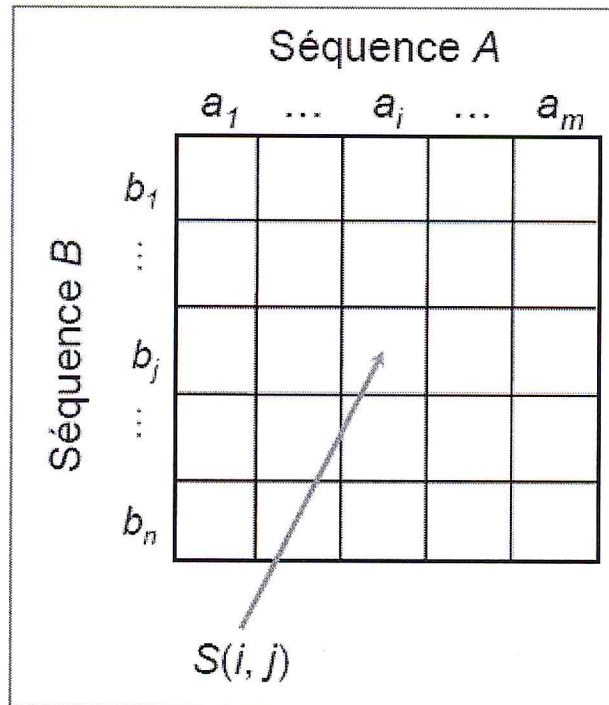


Fig.III.14 : La matrice des séquences.

Soit $S(i, j)$ la valeur optimum du score dans la case de coordonnées (i, j) par rapport aux scores issus des trois cases adjacentes $(i-1, j)$, $(i-1, j-1)$ et $(i, j-1)$ comme le montre la figure (Fig.III.15) :

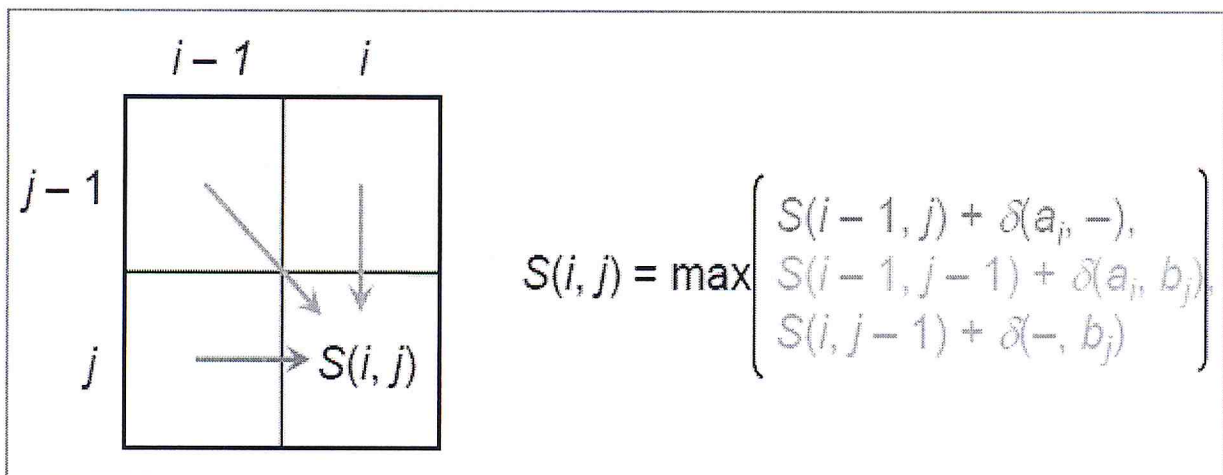


Fig.III.15: La méthode de calcul de la matrice de Needleman.

- **Bords de la matrice**

Les cases situées sur le bord du haut ou le bord gauche de la matrice (voir Fig.III.16) ne possèdent pas le total requis de trois cases précédentes, donc on ajoute une ligne et une colonne supplémentaires afin d'initialiser la matrice (initialisé la ligne et la colonne (0, n) et (m, 0) de la matrice).

Le calcul du score ne se faisant qu'avec des indices i et j supérieur à un.

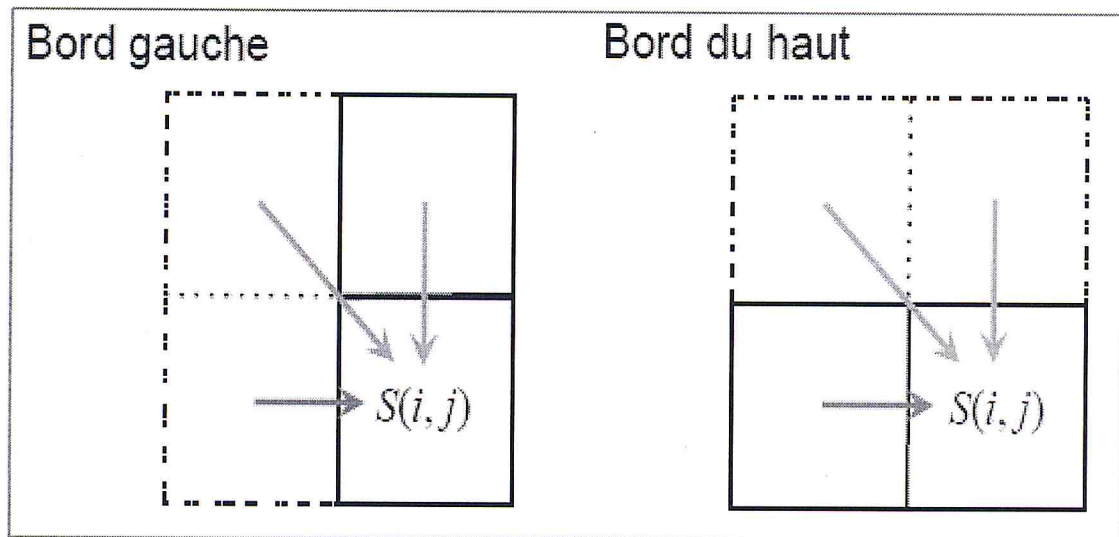
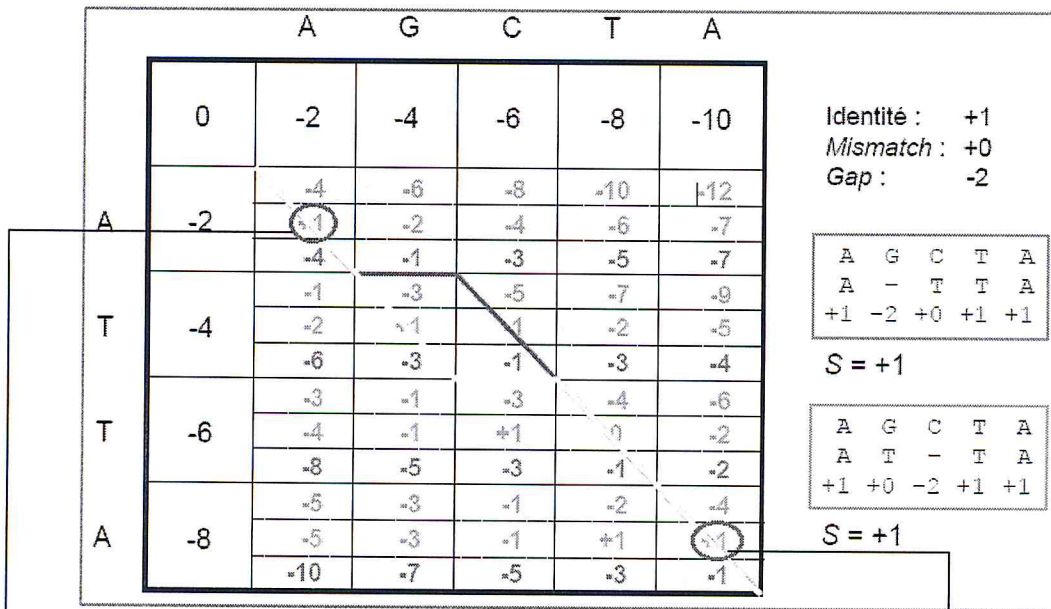


Fig.III.16: Bords de la matrice.

Dans le calcul du score de l'algorithme de Needleman, trois valeurs sont utilisées selon la similarité entre les deux textes. L'algorithme commence par aligner les deux séquences, une avec l'autre, si les caractères à la i ème position dans les deux séquences sont dissimilaires le caractère d'une séquence est aligner avec un vide ou « _ » (ce qui veut dire un Gap), s'ils sont similaires alors ils sont alignés l'une avec l'autre automatiquement (ce qui veut dire une Identité ou Match). Après la fin de l'alignement, les restes des séquences sont alignés l'une avec l'autre malgré qu'ils ne sont pas similaires (ce qui veut dire un Mismatch).

Depuis la partie précédente « Règles pour les scénarios » dans l'étape d'analyse, les attributs restants dans le fichier XML des scénarios (voir Fig.III.12); l'attribut « Compteur » définit la taille de l'échantillon des postures enregistrées en temps réel et l'attribut « Seuil » qui détermine si le résultat de cet algorithme est pris en considération en étant supérieur au seuil défini par l'utilisateur de l'application.

Exemple de calcul de la similarité en utilisant la méthode de Needleman-Wunsch :



Pour calculer la valeur de cette case, on doit additionner la valeur voisins gauche et haut avec le « gap » (ici c'est $(-2) + (-2) = -4$ et $(-2) + (-2) = -4$) et additionner la valeur du voisin en diagonal avec soit:

- * La valeur « match », si les deux caractères/chaines de caractères sont égaux (comme dans ce cas $G = G$, $0 + 1 = +1$)
- * Sinon la valeur « mismatch »

Puis choisir le maximum entre ces trois résultats donc:

$\text{Max}(-4, -4, +1) = +1$ et on garde la trace d'où vient ce résultat

S'il y a deux ou plus qui ont la valeur max, on garde leur 2 traces.

A la fin du calcul, on fait un retour vers l'arrière dépendamment des trace qui nous donnerons la distribution de la séquence B par rapport à A

Fig.III.17: Explication des calculs de la matrice de Needleman.

3. Conclusion :

Nous avons mis au point un système de reconnaissance de gestes qui nous a permis d'estimer à chaque instant à partir d'une séquence vidéo la position dans l'espace des différentes parties du corps, et de faire des traitements sur un geste acquis et de le comparer avec les gestes de la base des données déjà définie.

Nous avons aussi détaillé deux méthodes de calculs de similarité textuelles pour contribuer à l'estimation des similarités entre les postures enregistrées en temps réel par rapport aux scénarios définis.

Ce travail peut être vu comme la première étape d'un système plus global d'analyse du comportement, les résultats des tests effectués de ce chapitre seront enrichis en détails dans le chapitre suivant.

Chapitre IV: Implémentation et Résultats

1. Introduction :

Dans le cadre de notre projet qui consiste à développer un module de reconnaissance de gestes basée sur les interfaces Homme-Machine gestuelles qui sera inséré dans un système de vidéo protection.

Dans ce chapitre nous allons présenter les moyens utilisés pour la création et le développement de notre plateforme ainsi que la description de ses fonctionnalités.

La figure (Fig.IV.1) ci-dessous montre une représentation globale de notre application.

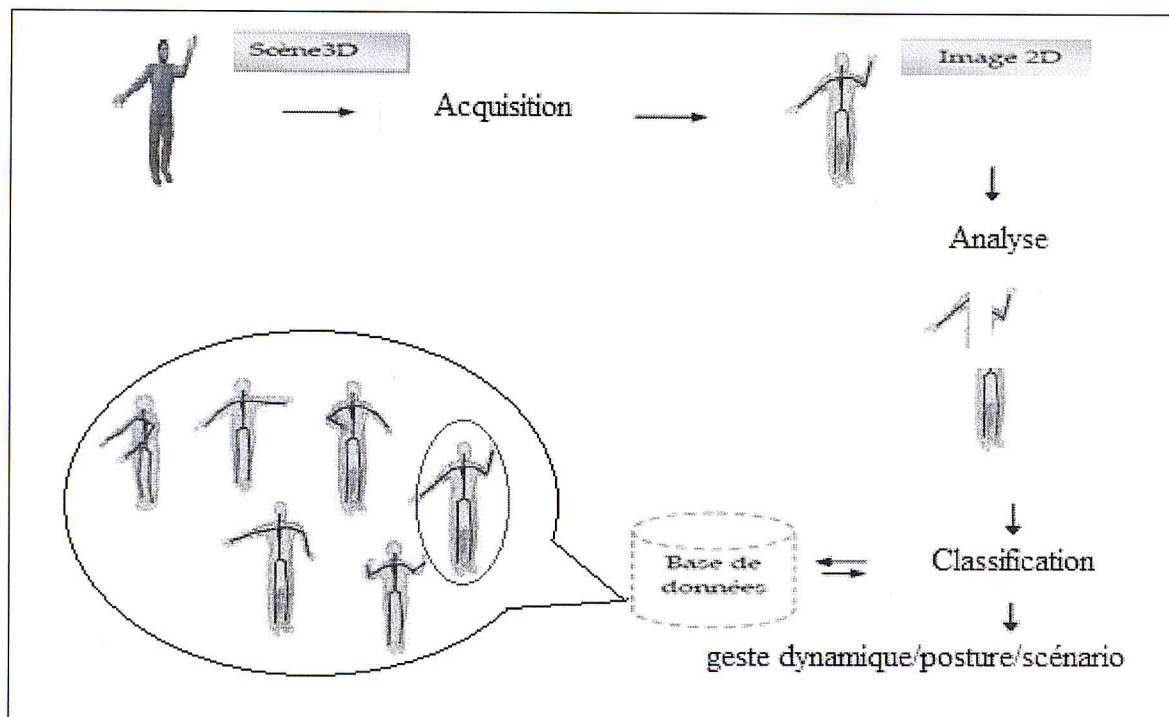


Fig.IV.1 : Schéma global du système développé.

2. Environnement de développement:

Avant tout commencement, il fallait choisir une solution de développement qui pourra nous aidez pour réaliser l'application. Nous avons opté pour l'utilisation du langage de programmation C# pour son type open-source [28]et le nombre important de ses utilisateurs dans le monde

2.1 Langage de programmationC# :

Le langage évolué C#(C sharp) est un langage de programmation orienté objet à typage fort, créé par la société Microsoft. Le C# est simple, puissant, de type sécurisé. C'est un nouveau langage de programmation qui a été conçu pour permettre la création d'une large gamme d'applications d'entreprise s'exécutant sur le .NET Framework[29].

Il a été créé afin que la plateforme Microsoft .NET soit dotée d'un langage permettant d'utiliser toutes ses capacités. Il est très proche du Java dont il reprend la syntaxe générale ainsi que les concepts (la syntaxe reste cependant relativement semblable à celle de langages tels que le C++ et le C). Toutefois, l'implémentation est plus proche de celle du Pascal Objet [29].

2.2 Microsoft .NET:

Le .NET Framework [40] est un modèle de programmation complet pour construire des applications mobiles, bureau et Web qui fonctionnent sous Windows

Le .NET Framework est désigné pour remplir les objectifs suivants [30]:

- Fournir un environnement cohérent de programmation orienté objet.
- Fournir un environnement d'exécution de code qui minimise le déploiement de logiciels et de conflits de contrôle de version.
- Fournir un environnement d'exécution de code qui favorise l'exécution sûre du code, y compris le code créé par une troisième partie inconnue ou moins confiant.
- Fournir un environnement d'exécution de code qui élimine les problèmes de performance des environnements scriptés ou interprétés.
- Pour rendre l'expérience de développeur cohérente pour tous les types variés d'applications, telles que les applications basées sur Windows et les applications basées sur le Web.
- Pour construire toutes les communications sur les normes de l'industrie pour veiller à ce que le code basé sur le .NET Framework peut intégrer à tout autre code.

2.3 Outils de développement Microsoft Visual Studio:

Nous avons choisi le logiciel **Microsoft Visual Studio** qu'est une suite de logiciels de développement pour Windows conçu par Microsoft

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Developer [31].



Fig.IV.2: Microsoft Visual Studio.

2.4 WinForms et WPF:

WinForms (Windows Forms) et WPF (Windows Presentation Foundation) sont des approches de Microsoft à un cadre de GUI (Graphical User Interface), utilisés avec le framework .NET [30].

Un cadre GUI permet de créer une application avec un large éventail d'éléments de l'interface graphique, tels que les étiquettes, textboxes et d'autres éléments bien connus et permet aux développeurs de se concentrer sur leurs applications [32].

Il y a beaucoup de cadres de GUI, mais pour les développeurs .NET, les plus intéressants sont actuellement WinForms et WPF. Microsoft a toujours maintenu le soutien de WinForms malgré que WPF soit le plus récent. Il y a quelques différences entre les deux cadres, mais leur but est le même: Pour faciliter la création d'applications avec une grande interface graphique [31].

La seule différence la plus importante entre WinForms et WPF est le fait que, bien que WinForms est simplement une couche au-dessus des commandes standards de Windows (par exemple, une zone de texte), WPF est construit à partir de zéro et ne repose pas sur les contrôles standards de Windows dans presque toutes les situations [32].

Nous avons choisi WinForms car [33]:

- Il est plus ancien et donc plus essayé et testé
- Il y a déjà beaucoup de contrôles de 3ème partie que vous pouvez acheter ou obtenir gratuitement.

- Le concepteur dans Visual Studio est toujours mieux pour WinForms que pour WPF.

Parmi les bibliothèques externes utilisées dans notre application :

- Microsoft. Kinect disponible dans le SDK (Kit de développement) Kinect v1.7
- Microsoft. Kinect. Toolkit disponible sur le site de Microsoft.com
- Input Simulator disponible sur le site codeplex.com

3. Présentation de l'application

Dans cette partie nous allons présenter les différentes interfaces et détailler leur principe de fonctionnement

3.1 Interface "fenêtre principale":

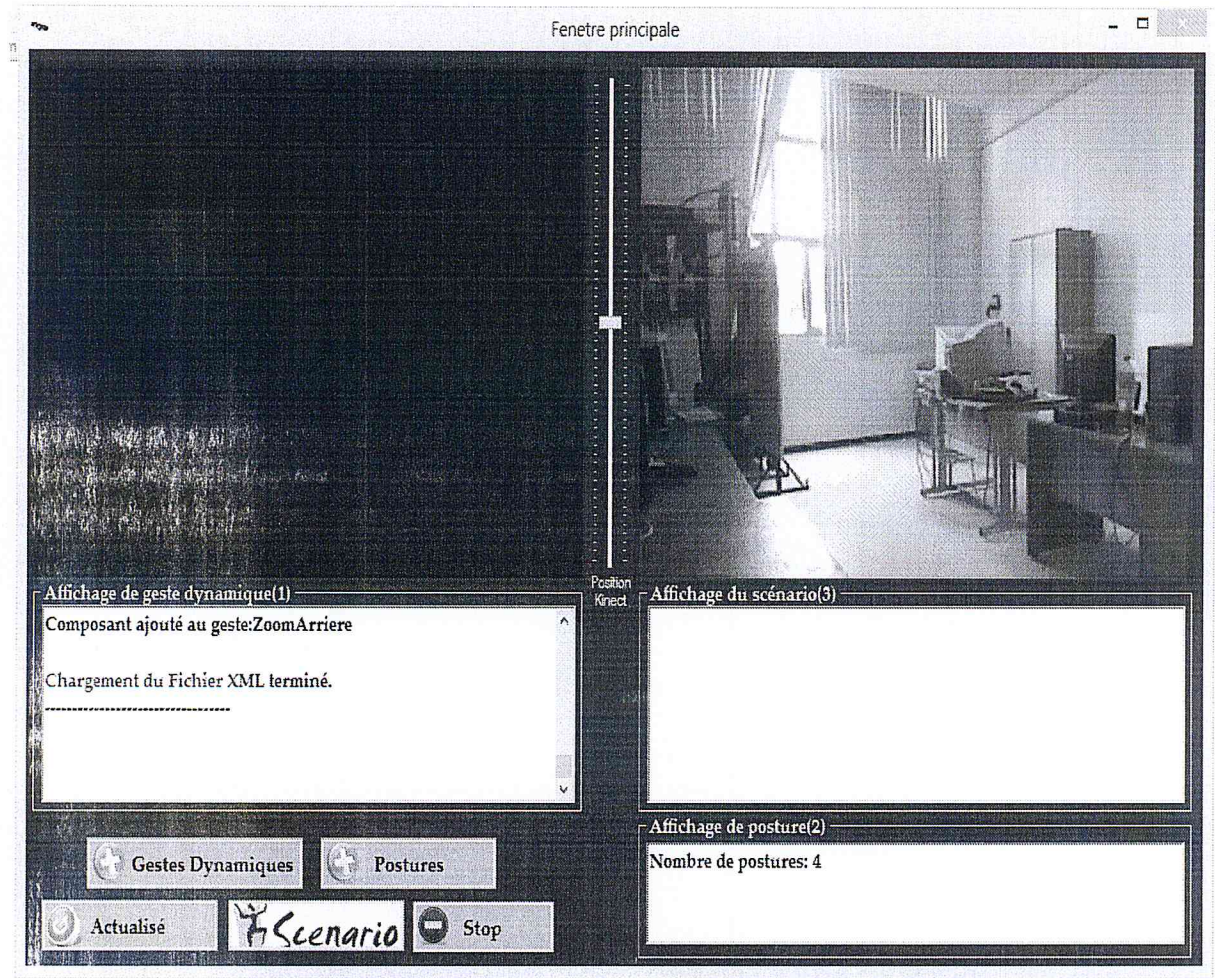


Fig.IV.3:Fenêtre principale.

Dans notre fenêtre principale, nous avons l'affichage de l'environnement surveillé en deux types, une au format RGB et l'autre montre seulement le squelette des individus détectés dans le champ de vue du capteur (voir Fig.IV.4).

Entre eux, nous avons un TrackBar qui permet de contrôler la position de la Kinect (augmenter ou diminuer son angle d'inclinaison).

Puis nous avons trois champs pour l'affichage des différents résultats des gestes de l'individu, le champ (1) qui montre les gestes dynamiques reconnus après traitement, (2) montre les postures reconnues après traitement et finalement (3) montre les différents scénarios et le résultat de l'algorithme de similarité.

Cinq boutons sont disponibles dont, trois d'entre eux nous permet d'accéder aux autres fenêtres et les deux autres: « Actualiser » qui permet de recharger la base de données du programme (Recharger le fichier XML) et le bouton « Stop » qui sert à arrêter le programme.

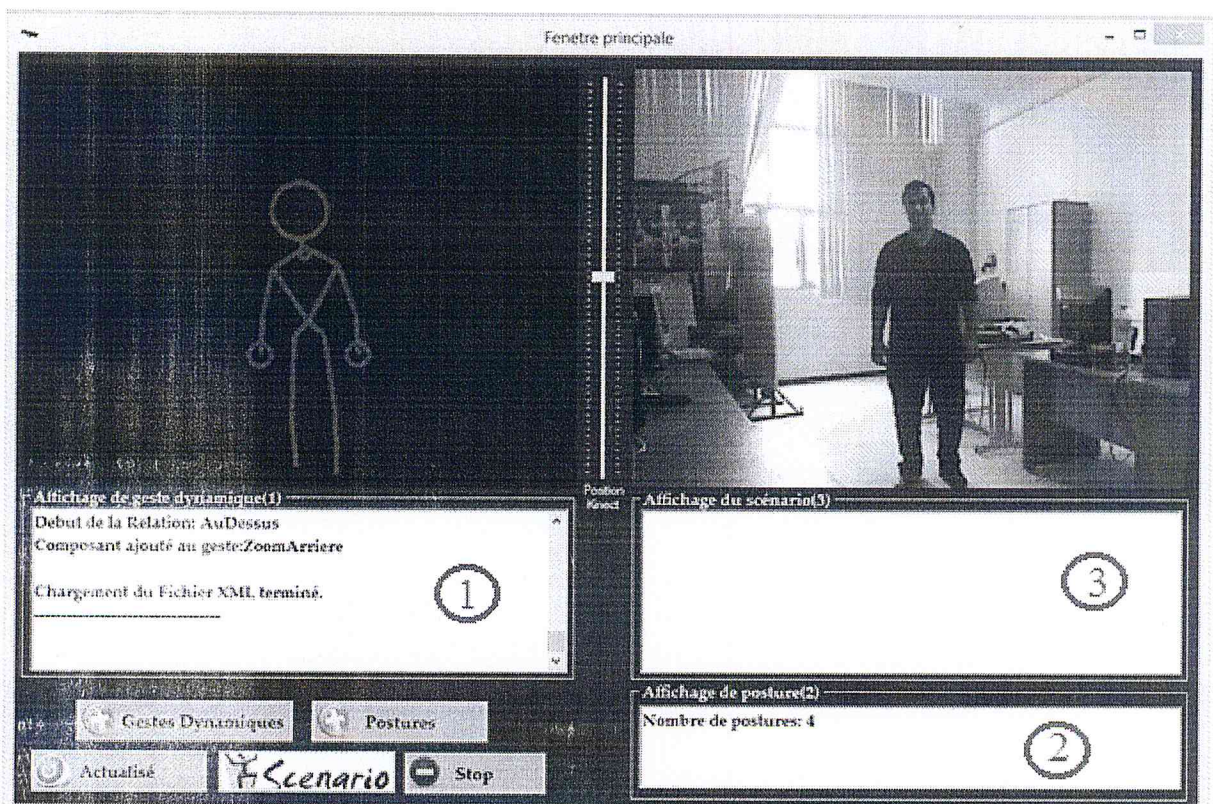


Fig.IV.4:Exemple qui montre l'affichage RGB et le squelette de l'individu présent dans le champ de vue du capteur.

3.2 Interface "nouveau geste dynamique":

Dans la figure(Fig.IV.5) suivante, nous montrons la fenêtre qui gère les gestes dynamiques :

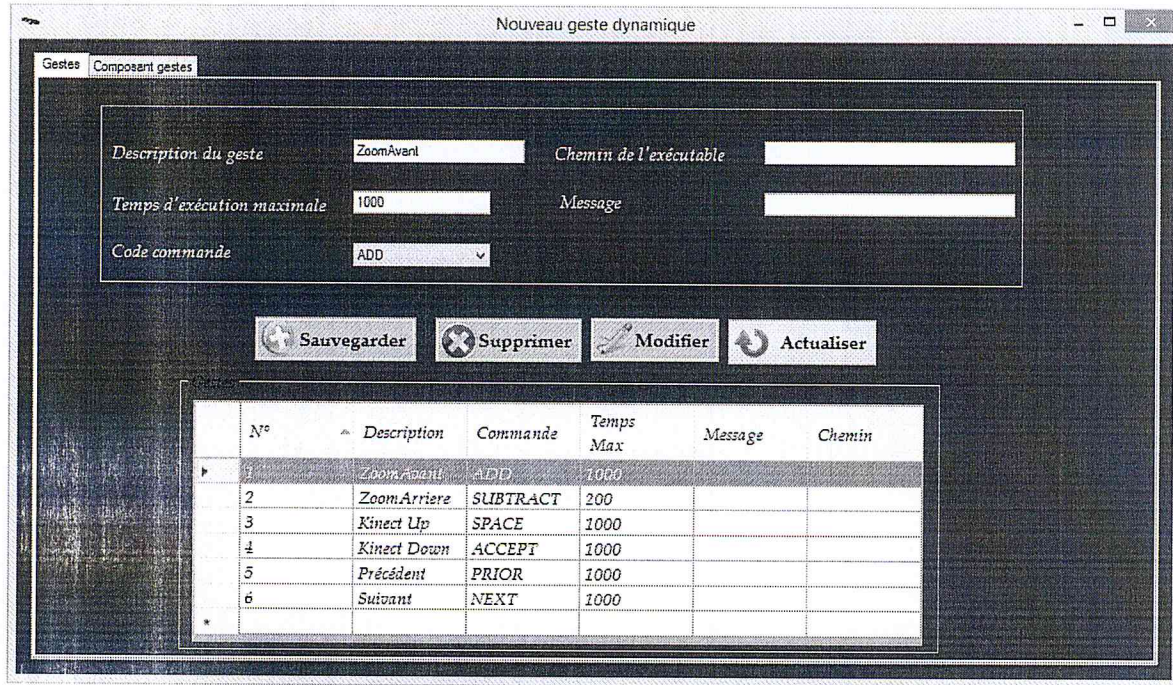


Fig.IV.5: Nouveau geste dynamique

Elle se constitue de deux onglets, un onglet pour les gestes que nous pouvons voir dans le tableau avec ses différents attributs (déjà insérés), et au-dessus de ce tableau nous avons les différents boutons et champs qui nous permettent de mettre à jour nos fichiers XML (base de données) soit en ajoutant un nouveau geste ou en modifiant un ou supprimer un geste. Et le bouton « Actualiser » pour actualiser cette fenêtre.

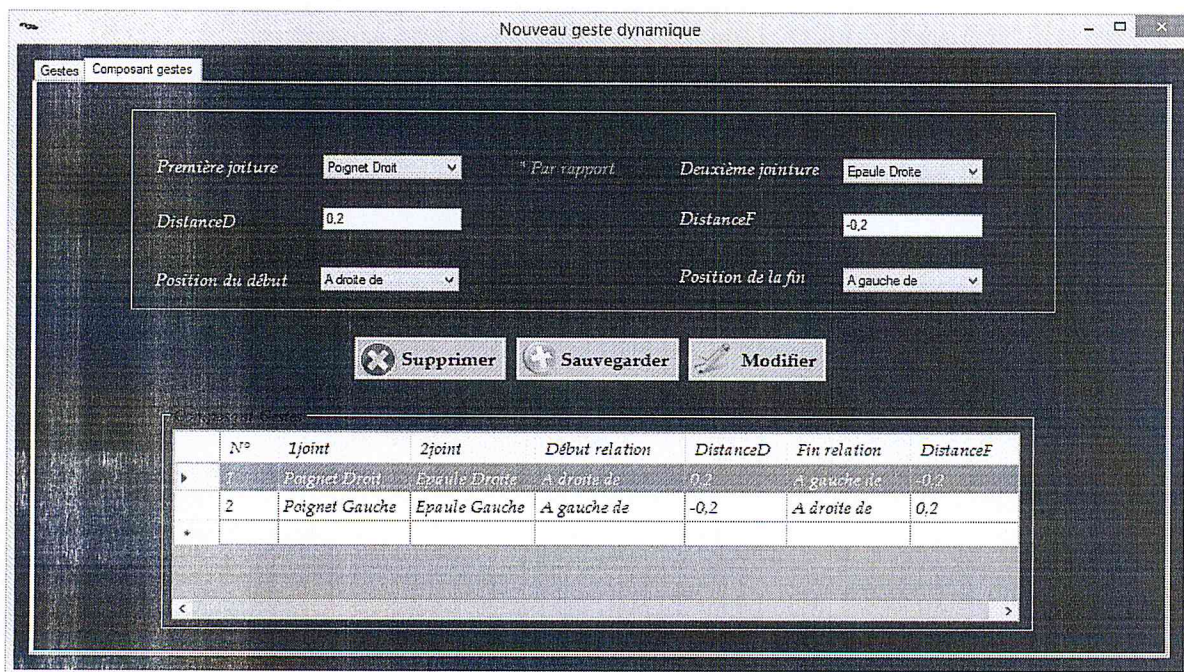


Fig.IV.6:Conditions du geste.

Et l'autre onglet (voir Fig.IV.6), pour remplir les conditions ou caractéristiques du geste. La même chose comme l'onglet précédent, un tableau pour l'affichage des conditions d'un geste sélectionné et des champs et boutons pour mettre à jour ces conditions. Un seul geste peut avoir une ou plusieurs conditions.

3.3 Interface "nouvelle posture" :

La figure suivante représente la fenêtre des postures :

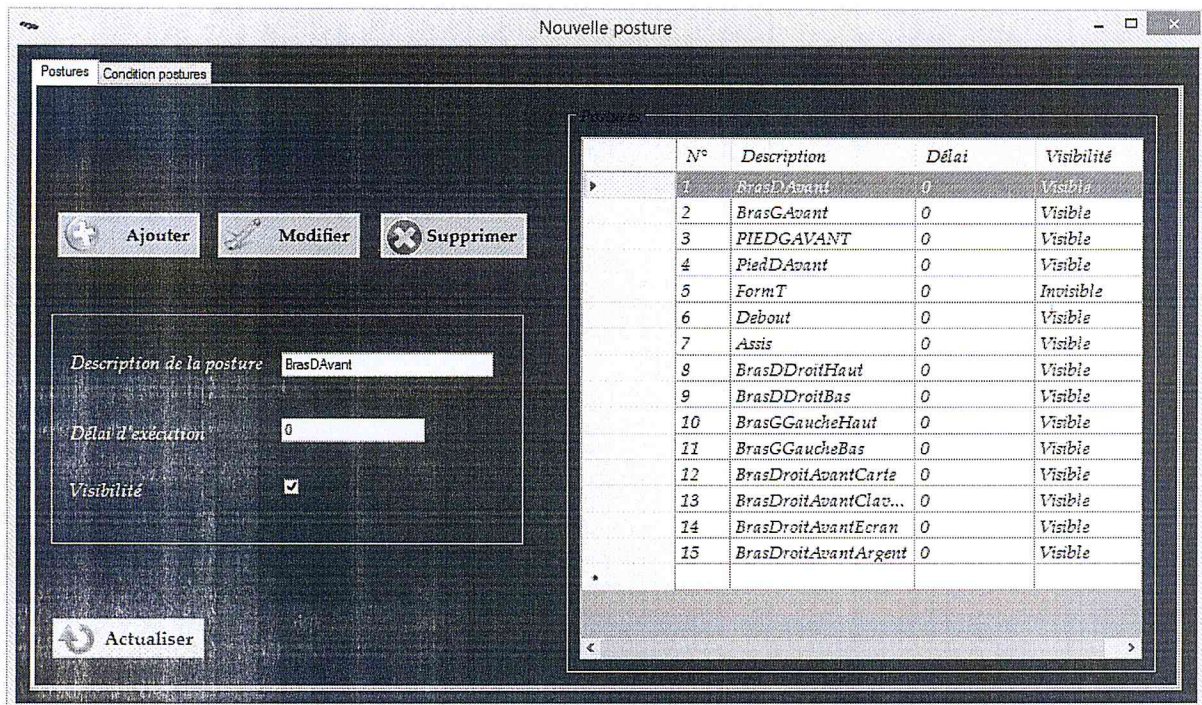


Fig.IV.7:Fenêtre "Nouvelle posture".

Comme la fenêtre précédente, nous avons ici deux onglets, une pour les postures et l'autre pour ses conditions. L'onglet des postures a un tableau pour lister toutes les postures disponibles ainsi que des champs et boutons à gauche pour gérer les postures.

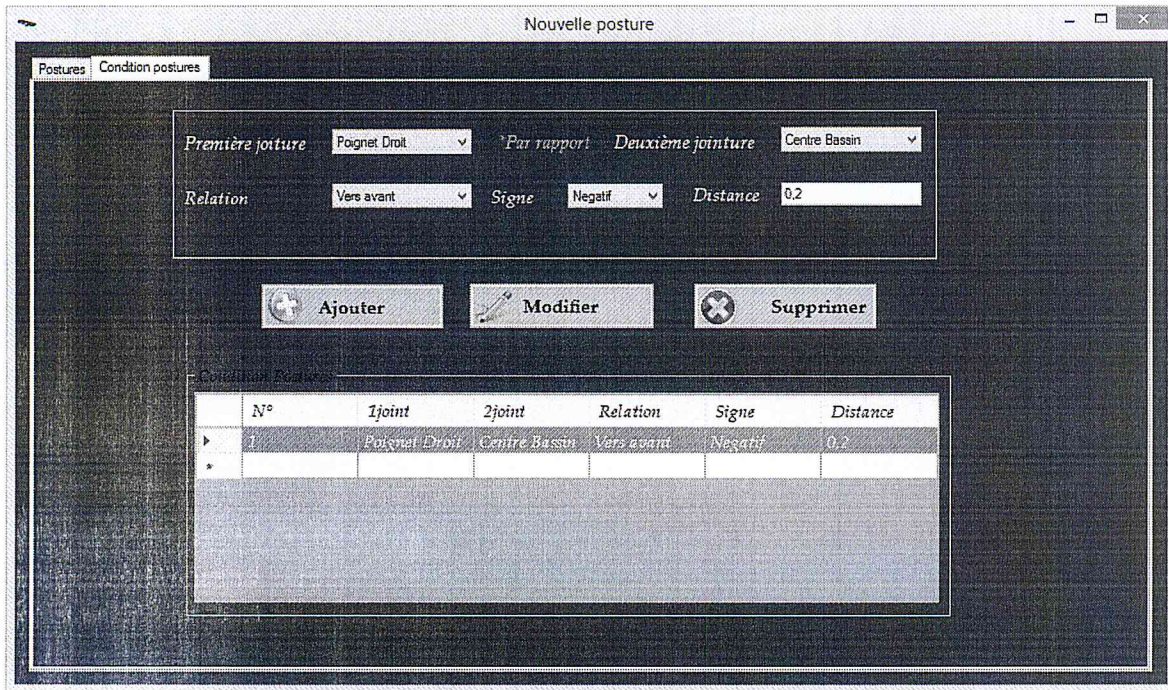


Fig.IV.8: Conditions/Caractéristiques des postures.

Et pour le deuxième onglet, nous avons la même chose que pour la figure (voir Fig.IV.6) du deuxième onglet pour les gestes dynamiques.

3.4 Interface "création scénario" :

Et dernièrement, nous avons ici la fenêtre des scénarios :

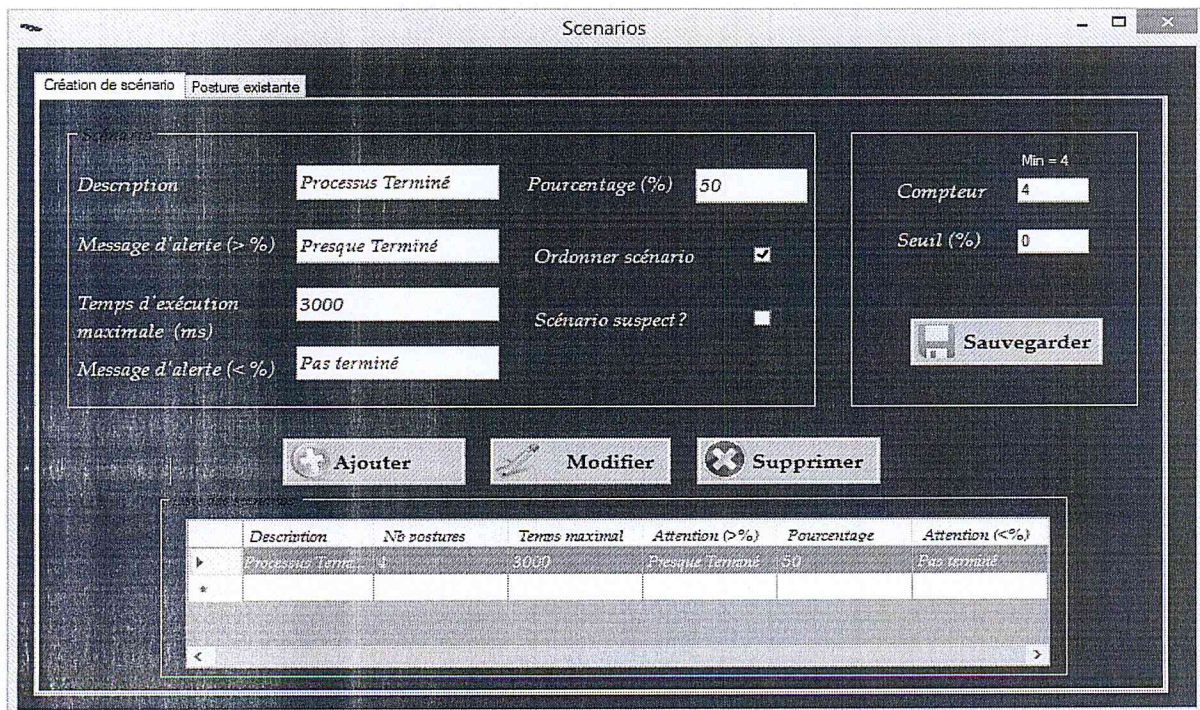


Fig.IV.9:Fenêtre "Création de scénario".

Un onglet où se trouve le tableau qui affiche la liste des scénarios dans le tableau (voir Fig.IV.9). Nous avons dans le côté gauche de la fenêtre les champs nécessaire pour gérer les scénarios. Et dans le côté gauche nous avons deux champs seulement, qui sont en relation avec l’algorithme de similarité utilisé (L’algorithme qui utilise les postures en temps réel et les scénarios enregistrées pour calculer leur similarité) pour déterminer le nombre d’enregistrement en temps réel avant le traitement et le seuil minimum pour prendre en considération ces résultats.

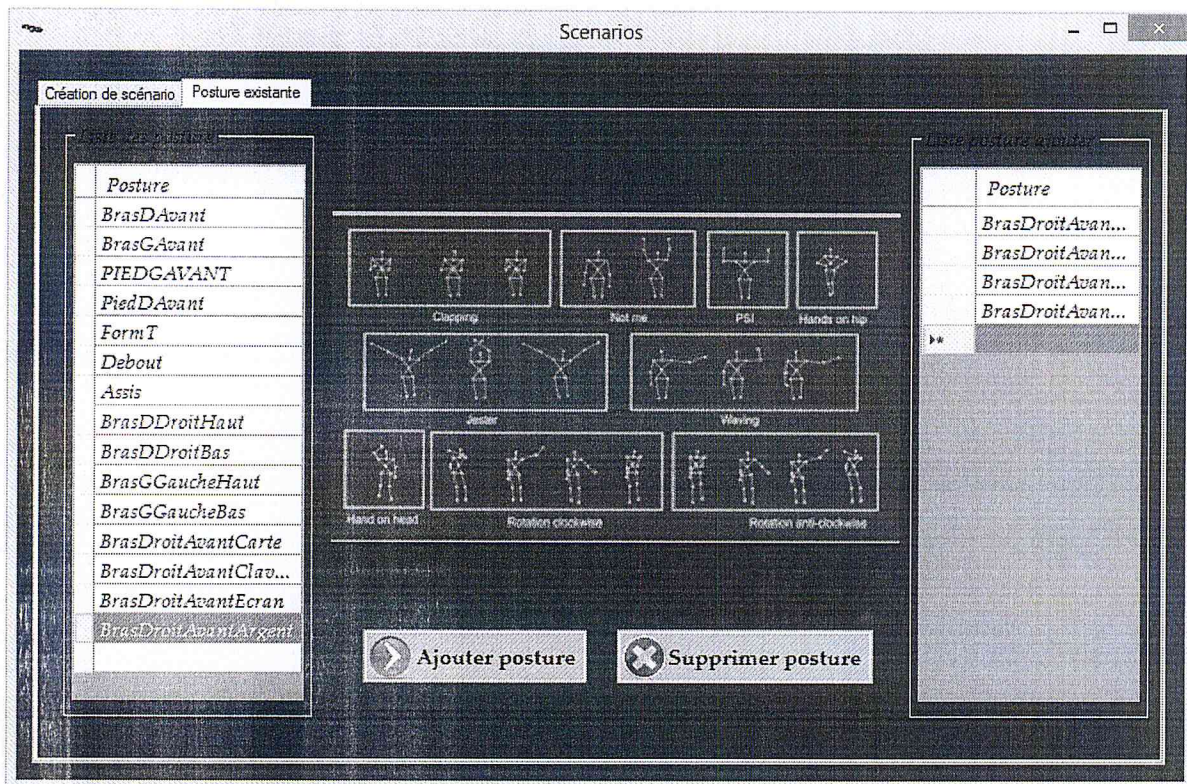


Fig.IV.10: Postures existantes pour la création d'un nouveau scénario.

Et dans cet onglet, nous avons deux tableaux : le premier sur la gauche (Liste des postures) représente tous les postures disponibles, et le second à droite (Liste posture à ajouter) qui fait deux tâches, il affiche les postures qui forment un scénario si un scénario est sélectionné, et, participe pendant l’ajout ou la modification des scénarios. Ce dernier peut être remplis grâce à la liste des postures disponibles en cliquant sur le bouton « Ajouter posture » ou enlever des postures en utilisant le bouton « Supprimer posture ».

4. Résultats des tests effectués

Dans cette partie nous allons montrer les différents résultats des tests effectués sur des cas réels, nous avons testé deux algorithmes développés, celui de la mesure de similarité "Cosinus" ainsi que celui de la mesure de similarité "Needleman-Wunsch". Nous allons commencer par la première mesure de similarité (cosinus) :

Voici les scénarios utilisés pour tester les deux mesures de similarité (voir Fig.IV.11)

```
<Scenarios Compteur="10" Seuil="50">
  <Scenario Description="Suspect" Attention="attSuspect" Pourcentage="50"
    Attention2="" TempsExecutionMax="5000" Suspect="True">
    <Partie Nom="BRASDAVANT" Ordre="1" />
    <Partie Nom="BRASGAVANT" Ordre="2" />
  </Scenario>

  <Scenario Description="Mouvement" Attention="attMouvement" Pourcentage="50"
    Attention2="" TempsExecutionMax="5000" Suspect="False">
    <Partie Nom="PDA" Ordre="0" />
    <Partie Nom="PGA" Ordre="0" />
    <Partie Nom="BRASGAVANT" Ordre="0" />
    <Partie Nom="BRASDAVANT" Ordre="0" />
  </Scenario>
</Scenarios>
```

Fig.IV.11: Scénarios utilisés pour tester les deux mesures de similarité.

Taux de reconnaissance des Scénarios avec un compteur =10.

Les scénarios :

Suspect = BDA BGA

Mouvement = PDA PGA BGA BDA

Tel que B = Bras, D = Droit = G = Gauche P = Pied A = Vers Avant

Test 1 :

Echantillon acquis par la Kinect : BDA BGA PDA PGA PDA PGA PDA PGA PDA PGA

Résultat :

Suspect : 24,2535625036333%

Mouvement : 85,74%

Test 2 :

Echantillon acquis par la Kinect : PDA PGA PDA PGA PDA PGA PDA PGA PDA PGA

Résultat :

Suspect : 0%

Mouvement : 70,71%

Test 3 :

Echantillon acquis par la Kinect : PDA PGA BDA BGA PDA PGA BDA BGA PDA PGA

Résultat :

Suspect : 60,30%

Mouvement : 95,94%

Nous remarquons, qu'après ces trois tests que la similarité cosinus compare le taux de répétitions des mots (postures) d'un scénario (plusieurs parties/postures) ou une partie de lui dans l'échantillon enregistré. Plus son contenu se répètent dans l'échantillon plus le pourcentage augmente tout en ignorant l'ordre des mots dans le scénario et prend en considération la répétition des mots malgré qu'ils ont été déjà trouvés dans l'échantillon.

Après avoir vu les résultats de la mesure de similarité en cosinus nous allons maintenant passer à la mesure de similarité Needleman-Wunsch, En utilisant les mêmes scénarios pour voir la différence entre les deux résultats des deux algorithmes.

Taux de reconnaissance des Scénarios avec un compteur =10.

Les mêmes scénarios des tests de similarité de cosinus :

Scénario1 = BD BG

Scénario2 = PD PG BG BD

Tel que B = Bras, D = Droit = G = Gauche P = Pied

Test 1 :

Echantillon acquis par la Kinect : BDA BGA PDA PGA PDA PGA PDA PGA PDA PGA

Résultat (La séquence A est l'échantillon enregistré en temps réel et la séquence B représente le scénario):

A ==>BDABGA PDA PGA PDA PGA PDA PGA PDA PGA

B ==> BDABGA / / /////

Similarité au scénario "Suspect" est de 100%

A ==>BDABGA PDA PGA PDA PGA PDA PGA PDA PGA

B ==>/ / / / / /PDA PGA BGA BDA

Similarité au scénario "Mouvement" est de 50%

Remarque : L'algorithme de Needleman met les mots de la séquence B non trouvés dans A à la fin en parallèle avec la séquence A.

Test 2 :

Echantillon acquis par la Kinect : PDA PGA PDA PGA PDA PGA PDA PGA PDA PGA

Résultat :

A ==> PDA PGA PDA PGA PDA PGA PDA PGA PDA PGA

B ==> / / / / / / / / BDA BGA

Similarité au scénario "Suspect" est de 0%

A ==> PDA PGA PDA PGA PDA PGA PDA PGA PDA PGA

B ==> / / / / / / / PDA PGA BGA BDA

Similarité au scénario "Mouvement" est de 50%

Test 3 :

Echantillon acquis par la Kinect : PDA PGA BDA BGA PDA PGA BDA BGA PDA PGA

Résultat :

A ==> PDA PGA BDA BGA PDA PGA BDA BGA PDA PGA

B ==> / / / / / / / BDA BGA / /

Similarité au scénario "Suspect" est de 100%

A ==> PDA PGA BDA BGA PDA PGA BDA BGA PDA PGA

B ==> PDA PGA / BGA / / BDA / / /

Similarité au scénario "Mouvement" est de 64,9375%

Nous avons apporté une amélioration à cet algorithme à savoir :

- Si les mots d'un scénario se trouvent complètement et successivement dans l'échantillon, donc le pourcentage est automatiquement 100% (en utilisant la méthode Contains).

Nous remarquons dans ces résultats que l'algorithme de Needleman parcourt l'échantillon pris en temps réel en le comparant avec les scénarios existants. Contrairement à la méthode cosinus, l'algorithme de Needleman prend en considération l'ordre des mots (postures) dans le scénario et écarte les mots déjà trouvé (évite la répétition).

Voici un tableau récapitulatif des résultats obtenus des tests précédents :

	Similarité Cosinus			Algorithme de Needleman		
	Test 1	Test 2	Test 3	Test 1	Test 2	Test 3
Suspect	24,25%	0%	60,30%	100%	0%	100%
Mouvement	85,74%	70,71%	95,94%	50%	50%	64,94%

Tab.IV.1 : Tableau comparatif des résultats obtenus des deux méthodes de similarité.

En prenant le seuil en tant que 50% (c'est-à-dire, la valeur est positif si elle est supérieure ou égale à 50% sinon négatif). Nous vérifions si le pourcentage est correct ou non :

	Similarité Cosinus			Algorithme de Needleman		
	Test 1	Test 2	Test 3	Test 1	Test 2	Test 3
Suspect	Faux négatif	Vrai négatif	Vrai positif	Vrai positif	Vrai négatif	Vrai positif
Mouvement	Vrai positif	Faut Positif	Vrai positif	Vrai positif	Vrai positif	Vrai positif

Tab.IV.2: Tableau interprétant la validité des résultats obtenus par rapport au seuil.

Après avoir fait ces différents tests nous avons optés pour le choix de la mesure de similarité de Needleman-Wunsch car elle nous donne de meilleurs résultats par rapport à la mesure de similarité cosinus et son algorithme prend l'ordre en considération contrairement a la mesure de similarité cosinus.

Jeu d'essai pour le distributeur de billet :

Notre application prend une approche vers un but un peu généralisé, c'est pourquoi, son utilité dépend de son domaine. Pour notre jeu d'essai, nous allons prendre l'exemple d'une personne qui veut retirer de l'argent d'un distributeur billet. Comment reconnaître ses intentions? Un simple retrait, vol ou acte de vandalisme? Pour cela :

- Seulement l'individu le plus proche est pris en considération.
- L'individu doit être près du distributeur avec une certaine distance fixe et connue (par exemple, 1,5 mètres).
- Si un individu reste sans bouger (une posture fixe) pendant un moment (par exemple, 30seconds), nous affichons à l'opérateur un message pour attirer son attention vers cet individu.
- Si un individu n'a pas terminé la procédure de retrait d'argent au bout d'un certain temps, un avertissement est envoyé.

A présent, voici notre base de données des postures utilisées (voir Fig.IV.12):

```

<Postures>

<Posture Description="BrasDroitAvantCarte" Visible="True" TempsAttenteExecution="0">
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" Relation="VersAvant" Distance="-0,2" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter" Relation="AuDessus" Distance="0,1" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" Relation="AGaucheDe" Distance="-0,05" />
</Posture>

<Posture Description="BrasDroitAvantClavier" Visible="True" TempsAttenteExecution="0">
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter" Relation="VersAvant" Distance="-0,2" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" Relation="AuDessous" Distance="-0,2" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter" Relation="ADroiteDe" Distance="0,2" />
</Posture>

<Posture Description="BrasDroitAvantEcran" Visible="True" TempsAttenteExecution="0">
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter" Relation="VersAvant" Distance="-0,2" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter" Relation="AuDessus" Distance="0,1" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" Relation="ADroiteDe" Distance="0,2" />
</Posture>

<Posture Description="BrasDroitAvantArgent" Visible="True" TempsAttenteExecution="0">
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" Relation="VersAvant" Distance="-0,2" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="ShoulderRight" Relation="AuDessous" Distance="-0,2" />
  <Condition PremiereJoint="WristRight" DeuxiemeJoint="HipCenter" Relation="AGaucheDe" Distance="0,2" />
</Posture>

</Postures>

```

Fig.IV.12: Postures utilisées pour tester le cas du distributeur de billets

Et voici notre scénario (voir Fig.IV.13) :

```

<Scenarios Compteur="4" Seuil="50">
  <Scenario Description="Processus Terminé" Attention="Presque Terminé" Pourcentage="40"
    Attention2="Pas terminé" TempsExecutionMax="30000" Suspect="False">
    <Partie Nom="BrasDroitAvantCarte" Ordre="1" />
    <Partie Nom="BrasDroitAvantClavier" Ordre="2" />
    <Partie Nom="BrasDroitAvantEcran" Ordre="3" />
    <Partie Nom="BrasDroitAvantArgent" Ordre="4" />
  </Scenario>
</Scenarios>

```

Fig.IV.13: Scénarios utilisé pour tester le cas du distributeur de billet.

Nous terminons ce chapitre par des captures d'écrans de l'application pendant l'exécution du scénario précédent (voir Fig.IV.14-17) :

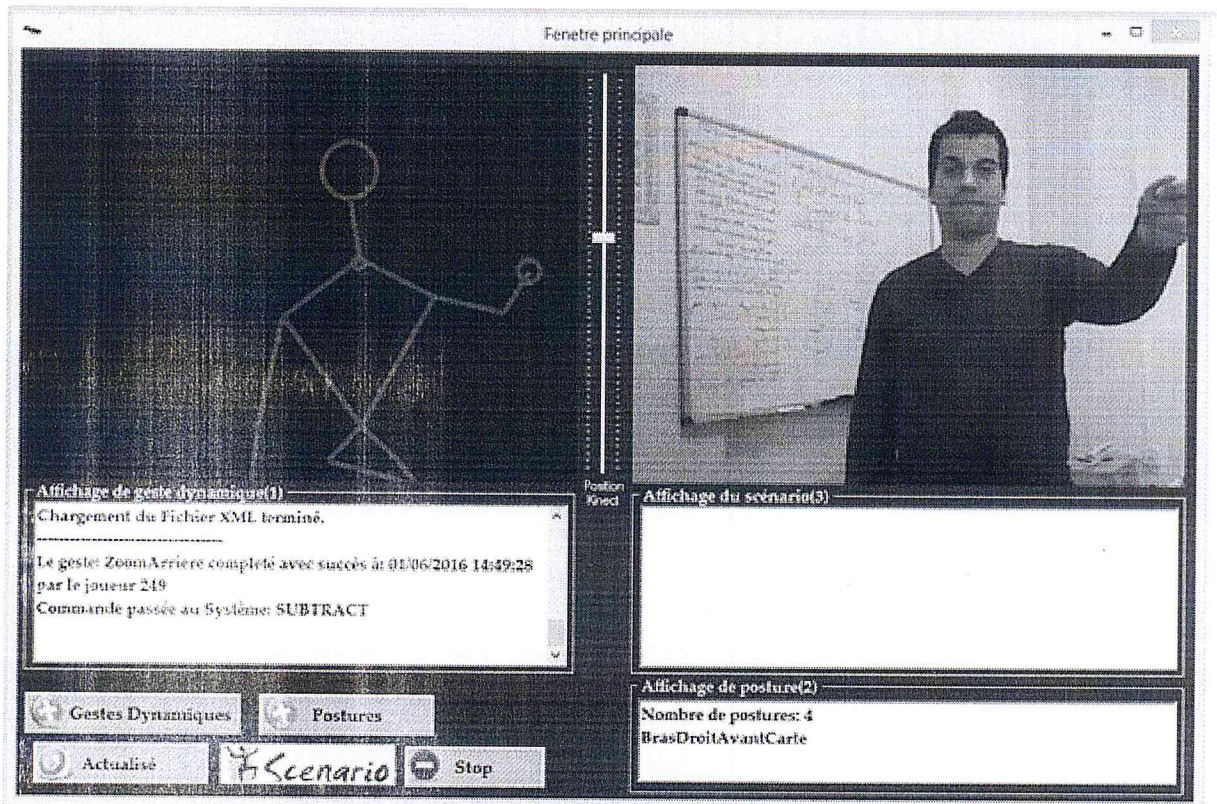


Fig.IV.14: Individu qui est entrain de mettre la carte dans le distributeur de billet.

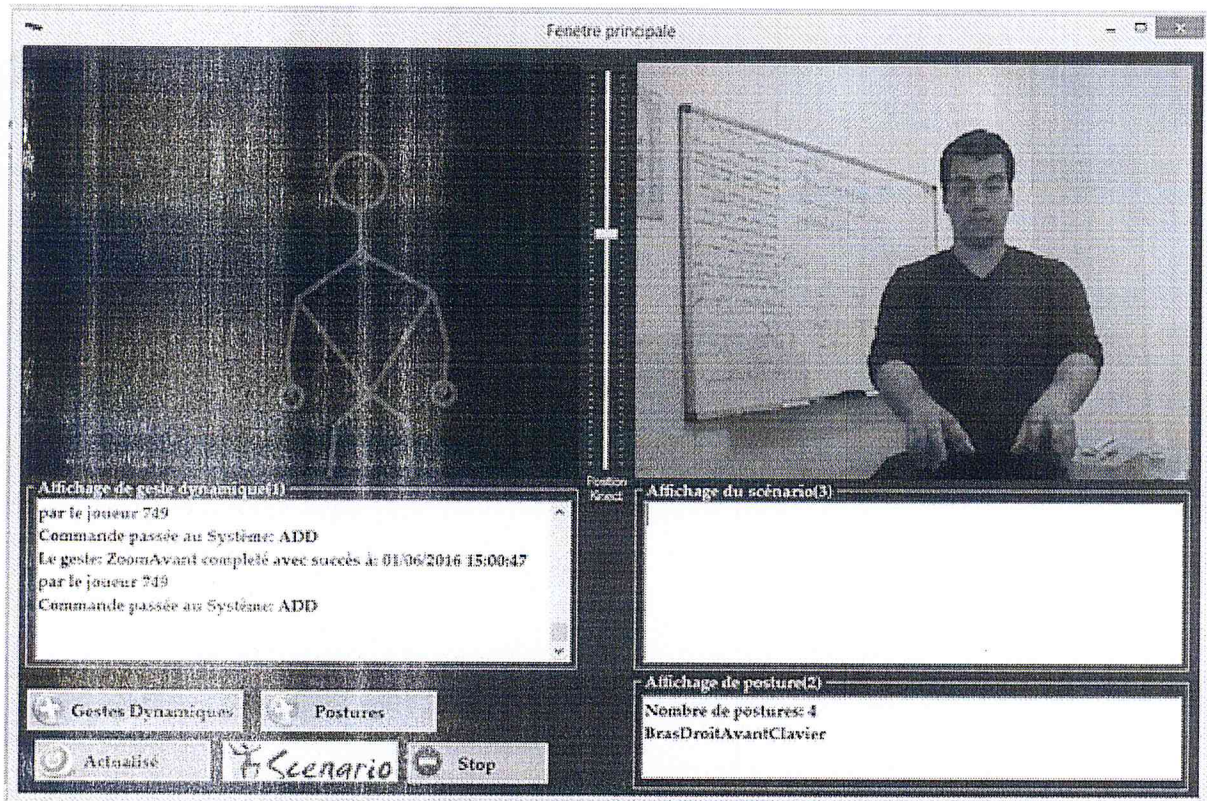


Fig.IV.15: Individu qui est entrain de saisir ces informations.

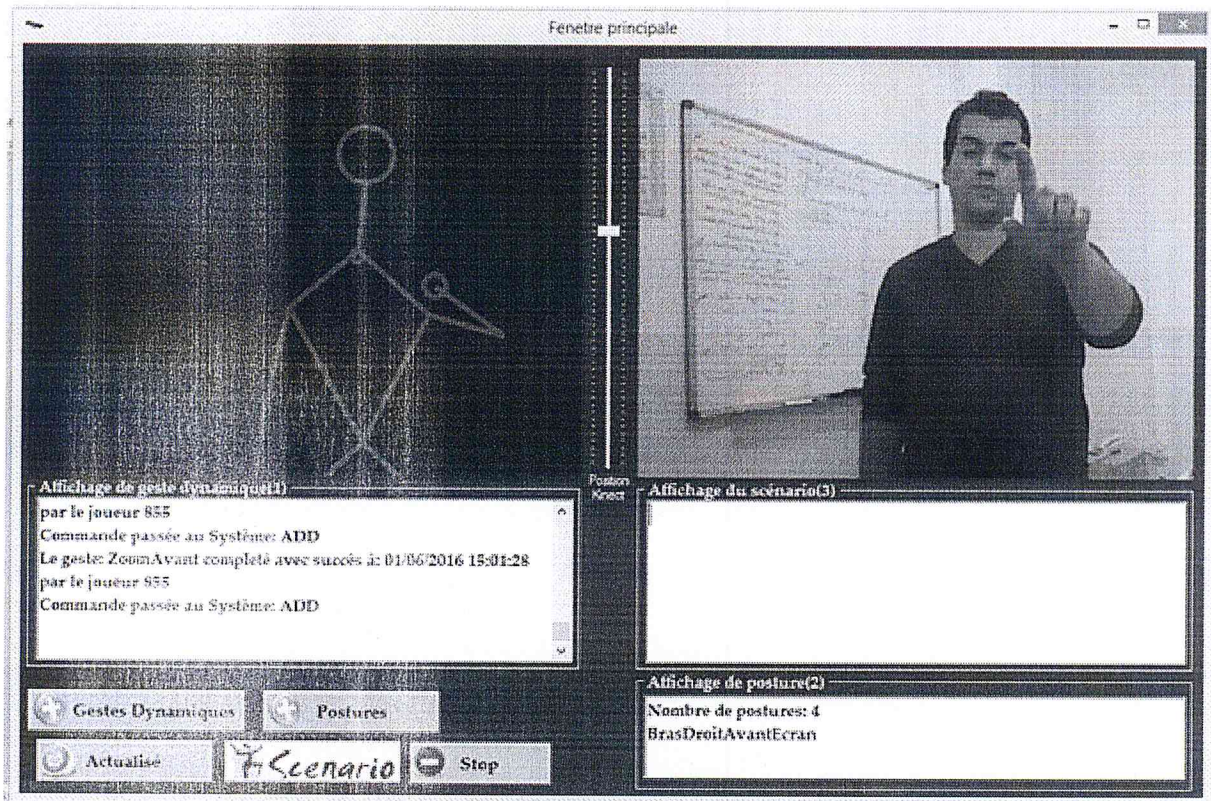


Fig.IV.16: Individu qui est entrain de sélectionner le montant d'argent.

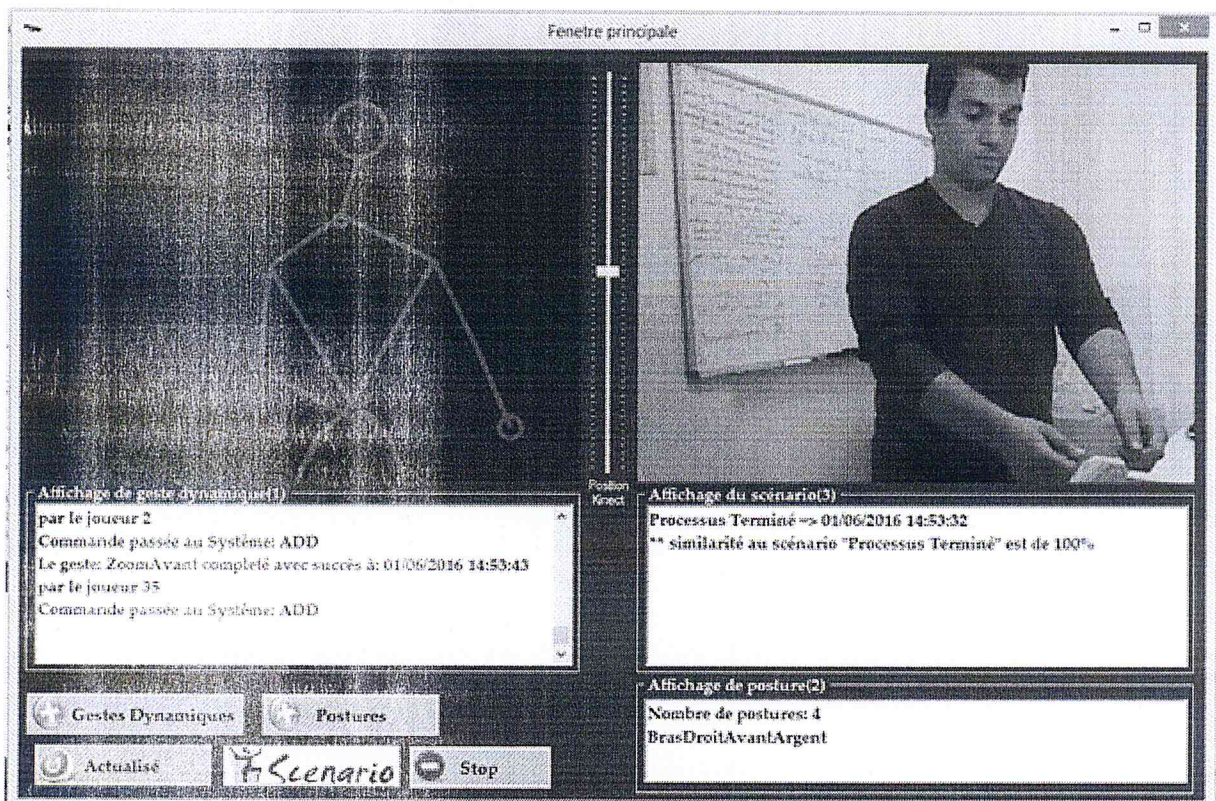


Fig.IV.17: Individu qui est entrain de retiré de l'argent.

Dans le cas où l'individu reste immobile pendant un moment, un avertissement sera affiché à l'opérateur (voir Fig.IV.18):

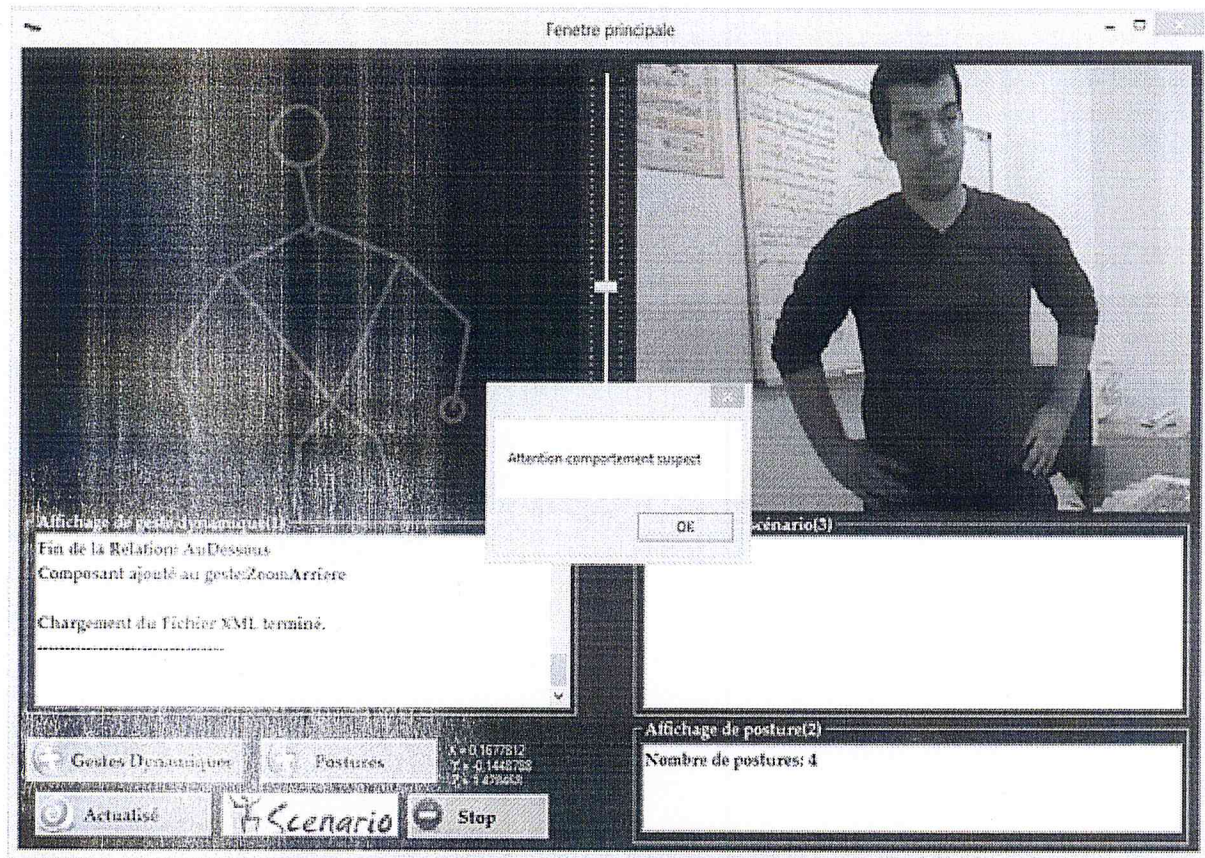


Fig.IV.18: Cas où aucun mouvement n'a été effectué.

5. Conclusion :

Nous avons montré - tout au long de ce chapitre – la méthodologie suivie pour la réalisation de notre module de reconnaissance de gestes humain. Nous avons montré aussi les différents outils utilisés pour la réalisation de notre module ainsi que l'environnement de développement.

De plus, nous avons montré et détaillé les principales interfaces que comporte notre module ainsi que leur mode d'utilisation.

Enfin nous avons terminé par les différents résultats récoltés à partir des différents tests réels réalisés durant l'établissement de cette application.

Les résultats obtenus montrent que notre module n'est pas fiable à 100% mais nous avons atteint nos objectifs de reconnaissance de geste à plus de 90%.

Conclusion générale

Depuis toujours, l'homme rêve de machines intelligentes, capables de dialoguer avec lui, de le comprendre, et de répondre aux questions qu'il s'est toujours posé. Dans cette quête la recherche en reconnaissance de gestes joue un rôle très important.

Pour cela, nous avons présenté un travail qui a pour but l'utilisation de la technologie Kinect comme capteur de vision pour détecter et reconnaître les gestes et postures suspect d'un individu. Celle-ci doit répondre à des exigences de rapidité et de robustesse des résultats pour concevoir et implémenter un ensemble d'algorithmes, sous forme de module, de reconnaissance de geste humain en temps réel afin de l'insérer dans un système existant de vidéoprotection.

La réalisation d'un tel module de reconnaissance de geste humain est donc le résultat de notre recherche et de notre implémentation. Ce résultat est le fruit de plusieurs tests de différentes méthodes, au cours des différentes phases de notre travail.

La période que nous avons passé pour le développement de ce projet de reconnaissance de geste nous a permis de découvrir plus profondément plusieurs aspects du développement d'une application complexe, de mieux comprendre la quantité et la structure du travail nécessaires pour un projet de recherche dans un domaine qui au début ne nous était pas familier. Il nous a fallu d'abord nous renseigner sur le côté algorithmique de la reconnaissance de gestes, et plus généralement de la vision par ordinateur, qui est un domaine en pleine expansion. Après Il nous a fallu résoudre plusieurs problèmes algorithmiques ayant plus ou moins un rapport avec les mathématiques; discipline importante dans le traitement d'images en général. Ces recherches nous ont donc menés à la réalisation d'un système de reconnaissance de gestes.

Après avoir recensé l'ensemble des informations dans le cadre de la reconnaissance de geste humain au niveau du CDTA (Centre de Développement des Technologies Avancées) et soulevé les problèmes existants au sein de celle-ci, nous sommes parvenus à proposer des solutions qui peuvent remédier à ces problèmes. Nous espérons qu'à travers ce travail nous avons pu aborder tous les points essentiels pour que notre solution trouve sa place dans le domaine de la reconnaissance de gestes, plus précisément, dans la vidéo-protection et qu'elle permette d'augmenter la bonne sécurité civile.

Nous pouvons affirmer que ce projet a été réellement bénéfique car nous avons appris un nouveau langage de programmation et de nouveaux concepts et technologies telle que la technologie Kinect. Aussi, nous avons pu faire preuve d'autonomie et de prise d'initiatives.

La notion de travail en équipe a aussi joué un grand rôle, nous avons en effet, privilégié la communication entre nous ainsi qu'avec notre encadreur et promoteur.

Pour conclure, il va sans dire que les fantasmes de l'homme actuel concernant sa relation avec les machines sont loin de devenir réalité, mais offrent d'innombrables opportunités pour la recherche en reconnaissance de gestes.

Limites et perspectives

Après avoir fini le développement de notre application, et au fur et à mesure des tests effectués, nous avons remarqué que pour définir des conditions très précises (tel que dans l'exemple dans le jeu d'essai dans le chapitre IV), modifier et tester les règles au fur et à mesure est nécessaire afin d'arriver à la définition des règles désirés. On peut constater que les résultats qui ne sont pas toujours les même à cause de quelques raisons comme celle en relation avec la forme du squelette généré, soit il est déformé (L'individu est dans les limites du cadre visuel de la Kinect ou à cause d'un obstacle entre la Kinect et l'individu), ou il n'est pas strictement face à la Kinect, ou si les zones générées dans l'espace à partir des règles sont croissantes, ou un certain membre du corps passe par une autre zone avant d'arriver à l'action intentionnée. Et ces mêmes raisons peuvent engendrer des résultats erronés de l'algorithme de mesure de similarité.

Ce qui nécessite des améliorations de notre système, telles que:

- Utilisation de la Kinect version 2 et mettre à jour le programme pour qu'il soit compatible avec cette version de Kinect.
- Améliorer nos règles de traitement du squelette pour avoir de meilleurs résultats de classification.
- Améliorer le programme afin qu'il puisse appliquer les règles selon la façade du squelette, puis mettre à jour les règles pour qu'ils ne dépendent pas seulement du point de vue de la Kinect mais de celui de l'individu aussi.
- Optimiser au maximum le temps de traitement si possible.

Bibliographie

Bibliographie

Code	Reference
[1]	M. Elarbi-Boudiher, Khalid A. Al-Shalfan, Artificial Inteligence Lab, Imam University, Riyadh. KSA, 2009.
[2]	M. Ferjani, A. Serial, H. hennani, A. Seddik, Vidéo Surveillance à Distance, Rapport de TER MASTER1 IFPRU, 2009.
[3]	Technologue pro, EVOLUTION DES SYSTEMES DE VIDEOSURVEILLANCE, Lien: http://www.technologuepro.com/cours-videosurveillance/Chapitre-3-evolution-des-systemes-de-videosurveillance , 2001.
[4]	COMPAOERE H. Hubert DeLorenzo et SAWADOGO W. Abdel-Aziz, Etude des systèmes de Vidéosurveillance, 2006-2007.
[5]	AIT ALIOUA Hamouche et SADOUD Mohamed, Vidéosurveillance intelligente : Plateforme de gestion et de traitement vidéo d'un réseau de caméras IP, École nationale supérieure d'informatique (esi), 2013/2014.
[6]	AFRICOM GROUP, Catalogue pour matériel de vidéo surveillance, lien : http://groupe-africom.com/images/groupe-africom/telechargement/video-surveillance.pdf , 2003.
[7]	Aude-Emmanuelle fleurant, La vidéosurveillance intelligente : promesse et défis, Rapport de veille technologique et commerciale, CRIM Technopole Défense et Sécurité, 8 avril 2009.
[8]	S. Boukir, F. Chenevière, 'Conception d'un système de reconnaissance de gestes dansés', Université de La Rochelle, 2 septembre 2003.
[9]	Julien Thomet, "Une vue d'ensemble de la reconnaissance de gestes", Département d'informatique Université de Fribourg.
[10]	BERRACHED Chahrazed, 'Système de reconnaissance de gestes de la main', 2013-2014.
[11]	Maria Karam and m. c. schraefel. 'Taxonomy of gestures in human computer interactions', 2005.

[12]	Bolt, R. A. Put-that-there: Voice and gesture at the graphics interface. In Proceedings of the 7th annual conference on Computer graphics and interactive techniques. ACM Press, 1980.
[13]	H.BENSENANE, A.HAMZA « Asservissement Gestuel avec apprentissage par réseau de neurones », USTO, LAERSI, PFE, ELN, 2008-2009
[14]	BARDIAU Romain et al, Développement d'une API de reconnaissance de gestes pour le capteur KINECT.
[15]	Jean-François Jego. Interaction basée sur des gestes définis par l'utilisateur : Application a la réalité virtuelle. Other. Ecole Nationale Supérieure des Mines de Paris, 2013.
[16]	Najeh NAFFAKHI Apprentissage supervisé pour la classification des images à l'aide de l'algèbre P-tree, Université de Tunis Institut Supérieur de Gestion de Tunis, Février 2004.
[17]	Ould Ahmed, Val khyarkoum, Classification des régresseurs non linéaires en utilisant un classificateur SVM, Université hadj lakhdar Batna Faculté de Technologie Département d'électronique, 2012.
[18]	V. N. Vapnik, "The nature of statistical learning theory". New York, NY, USA: Springer-Verlag New York, Inc., 1995.
[19]	SushmitaMitra and TinkuAcharya. Gesturerecognition: A survey. IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS - PART C, 37(3):311{324}, 2007.
[20]	Thomas B. et al. A survey of computer vision-based human motion capture. Comput. Vis. Image Underst, 2001.
[21]	Ludovic ROLAND, Structurez vos données avec XML, Le 20/12/2012.
[22]	Maxime Devanne, Modélisation 3D d'un corps humain à partir d'une caméra 3D Kinect, 2012.
[23]	Greg Borenstein, Making Things See 3D Vision with Kinect, Processing, Arduino, and MakerBot, 2012.
[24]	Mathieu Barnachon, Reconnaissance d'actions en temps réel à partir d'exemples, 22 avril 2013.

[25]	Martin Rajman, Ludovic Lebart, similarité pour données textuelles, 1998
[26]	Micro Joe, Similarité cosinus et recherche textuelle, lien: http://pdp.microjoe.org/tutoriels/258/similarite-cosinus-et-recherche-textuelle , 2013-2015.
[27]	NEEDLEMAN S. B. & WUNSCH C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48(3), 443–453, 1970.
[28]	Jeff Martin, Le compilateur C# délivré en Open Source, lien: https://www.infoq.com/fr/news/2014/04/roslyn_oss , 25 avr. 2014
[29]	Bertrand Cottenceau, C# / .NET support de cours, ISTIA, 2012-2013
[30]	Microsoft, Overview of the .NET Framework lien : https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx , 2016
[31]	Microsoft, lien: https://msdn.microsoft.com/fr-fr/library/fx6bk1f4(v=vs.90).aspx , November 2007.
[32]	Dwight Miller, windows presentation foundation vs winforms, lien: http://www.infragistics.com/community/blogs/devtoolsguy/archive/2015/04/17/windows-presentation-foundation-vs-winforms.aspx , 2015.
[33]	wpf-vs-winforms, Lien: http://www.wpf-tutorial.com/about-wpf/wpf-vs-winforms/ , 2016.