

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab Blida 1

N° D'ordre :



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Baraka Larbi Larid Islam

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : génie des systèmes informatiques

Sujet : Classification de personnes pour la robotique de service

Soutenu le : 26 / 06/2016

Devant le jury :

M Chikhi Nasim

Président

M. Zair

Examineur

M. Abdallah Kameche

Promoteur

Mme. Souhila Kahlouche

Encadrante

**Promotion
2015/2016**

DEDICACES

Au tout puissant

*Qui m'a donné la foi et de m'avoir permis d'en arriver
là ;*

A ma mère

*Ton amour pour nous, a laissé en moi une empreinte
indélébile. Que le seigneur nous accorde la grâce de
t'avoir encore à nos côtés pour longtemps afin que tu
sois témoin de notre reconnaissance ;*

A mon cher père

*Je te prie de trouver en ce travail le fruit de tes
efforts pour mon éducation et de vivre assez
longtemps pour en jouir ;*

A mon frère Ahmed

*A mon binôme qui a partagé avec moi les moments les
plus durs et les plus beaux de ce voyage
Islam ;*

A mes amis Bilal, Mounir, Abdessalam.

Je dédie ce modeste travail.

Larbi

DEDICACES

A MES TRES CHERES PARENTS

A MES SŒURS

ET A TOUS CEUX QUI SOUTIENT LE DEVELOPPEMENT INFORMATIQUE

Remerciement :

Ce travail a été effectué sous la direction de Monsieur **ABDALLAH KAMECHE**, Qu'il veuille bien trouver ici nos respectueux remerciements pour la confiance qu'il nous a accordée pour cette étude, et de la bienveillante attention qu'il nous a accordé à ce travail tout au long de son déroulement.

On remercie madame **SOUHILA KAHLUCHE** Professeur à l'institut **CDTA** de nous honorer en nous soumettant cette intéressante étude, pour la solitudo, les précieux conseils, son aide, l'intérêt constant qu'elle nous a accordée.

On remercie le **PRESIDENT DU JURY**, et on est heureux de pouvoir lui exprimer ici notre profonde reconnaissance.

Que les **MEMBRES DE JURY** trouvent ici l'expression de notre profonde gratitude pour avoir accepté de juger ce travail.

Que tous ceux qui ont contribué à la réalisation de cette étude de près ou de loin qu'ils veuillent bien accepter nos remerciements, en particulier Madame **OUKID** pour son aide, et tous les camarade de promotion.

Sommaire:

Chapitre 1: introduction à la robotique

1. Introduction :	1
2. Définition et Historique :	1
2.1. Définition de la robotique :	1
2.2. Histoire de la robotique :	1
Les automates :	2
Les Robots sans IA :	2
Les Robots avec IA :	3
3. Classification Des Robots :	4
3.1. Les manipulateurs :	4
3.2. Les télémanipulateurs : appareils de manipulation à distance (pelle mécanique, pont roulant), apparus vers 1945 aux USA :	4
3.3. Les robots :	4
4. Types de Robots :	4
4.1. Les robots Médical :	4
4.2. Les robots divertissements :	5
4.3. Les robots Militaires :	5
4.4. Les robots Domestique :	5
4.5. Les robots Industriels :	6
4.6. Les robots de Services :	6
Définition :	7
Catégories de la robotique de services :	7
LES DOMAINES APPLICATIFS :	13
4.7. Périmètre D'étude :	16
5. Middleware robot :	17
5.1. Définition :	17
5.2. Existant robot de middleware :	17
5.3. ROS :	18
Définition ROS :	18
Apport de l'outil ROS :	18
Principe de conception de ROS :	19
Système de Fichiers ROS :	20
Système de traitement et de calcule ROS :	21
6. Conclusion :	23
1. Introduction :	24

2. La Détection d'objet et les approches de suivi :	24
2.1. Détection Orientée objet :	24
2.2. Approches base sur la forme :	25
2.3. Approches base sur la couleur :	25
2.4. Détection basé sur la correspondance de modèle :	25
2.4.1 Fixe :	26
2.4.2 Déformable :	26
3. Détection de Mouvement :	27
3.1. Détection de mouvement par segmentation :	27
3.1.1 Détection par changement d'image :	28
4. Le suivi d'objet en mouvement :	28
4.1. Les approches basent sur la frontière :	29
4.2. Les approches basées sur la région :	29
5.2. Techniques utilisées dans la « mocap » :	30
5.3La Kinect :	34
5.3.1Les caractéristiques de la Kinect :	34
5.3.2La capture de la profondeur :	36
5.3.3Utilisation de la Kinect :	36
5.3.4La détection et le suivi en utilisant la Kinect :	38
Méthodologie :	39
6. Conclusion :	42
1. Introduction :	43
2. Définition :	43
2.1. Classification Supervisée :	43
2.2. Classification Non-supervisée :	44
3. Indices de ressemblance :	44
3.1. La distance euclidienne :	44
3.2. La distance de Manhattan :	45
3.3. La distance de Minkowski :	45
3.4. La distance Canberra :	45
4. Algorithmes de classification :	46
5. Conclusion :	49
1. Introduction	50
2. Conception :	51
2.1. Acquisition des données :	51
Outils utilisables :	53

TRAVAIL SUR L'IMAGE EN PROFONDEUR :	53
2.2. Détection & suivi de l'être humain :	55
2.3. Extraction des données :	57
Publication :	57
Réception :	58
2.4. La classification :	59
3. Conclusion :	66
1. Introduction :	67
2. Architecture globale :	67
3. Architecture de l'application :	68
4. Environnement de Développement :	68
5. Package ROS :	69
5.1. Définition d'un Package ROS :	69
5.2. Création d'un nouveau package sous ROS :	69
6. Tests des fonctionnalités :	70
6.1. Tests effectués en simulation :	70
Test pour une seule personne :	70
Test pour deux personnes :	71
Test pour trois personnes :	72
Test pour plus de trois personnes :	73
6.2. Tests effectués sur la plate-forme robotique expérimentale réelle :	74
7. Performances :	76
8. Fiabilité :	77
9. Conclusion :	77

Table des Figures :

Figure 1 : Le canard mécanique de Jacques De Vaucanson	2
Figure 2: Grey Walter et une tortue cybernétique sans carapace	3
Figure 3 : Wabot-1	3
Figure 4 : un robot médical	4
Figure 5 : un robot de divertissement	5

Figure 6 : Robot militaire	5
Figure 7 : Excmple Robots maison.....	6
Figure 8 : Exemple Robots Industriels	6
Figure 9 : Robot d’inspection de surveillance Jazz Security	9
Figure 10 : Robot d’assistance au geste (ou cobots) RB3D.....	10
Figure 11 : Robot thérapcutique Paro.....	11
Figure 12 : Robot de déminage Packbot 510 EOD.....	14
Figure 13 : Robot d’inspection et réparation de canalisation KASRO 4.0 DN200-DN600	15
Figure 14 : fonctionnement du système ROS	23
Figure 15 : Les marqueurs sous forme de boules et les camcras envoi.....	31
Figure 16 : Exemple réel de Andy-Serkis alias Gollum dans le Seigneur des Anneaux	31
Figure 17 : Les capteurs led placé sus le corps de l’acteur	32
Figure 18 : Exemple montre le processus pour crée un avatar en quatre étapes.	32
Figure 19 : image montre les principales parties de la Kinect (image face).....	34
Figure 20 : champ de vision de la kinect	35
Figure 21 : Architecture du driver Open Ni.....	36
Figure 22 : la différence entre openni, sdk windows et MoCap.....	37
Figure 23 : l'architcture globalc du suivi d’A.kar.....	39
Figure 24 : montre la segmentation de la peau (la peau en couleur blanc)	40
Figure 25 : les mesures du corps humain.....	41
Figure 26 : les points du squelette de l’être humain	41
Figure 27 : Excmple d’une hiérarchic	46
Figure 28 : Exemple de deux classes avec deux critères.....	47
Figure 29 : différence entre image simple et image profondeur	52
Figure 30 : exemple d'une DepthMap.....	53
Figure 31 : excmple d'une DcpthMap organisé	54
Figure 32 : les coordonnées x,y,z	55
Figure 33 : la position 'PSI'	56
Figure 34 : les différents composants pour publier un message.....	58
Figure 35 : les différents composants pour la réception d'un message.....	58
Figure 36 : Fonctionnement de l’approche	59
Figure 37 : pseudo algorithme de la fonction principale.....	60
Figure 38 : Enregistrement des coordonnées	61
Figure 39 : le processus de la 1 ^{ere} étape.....	62
Figure 40 : Calcul des couples de personnes les plus proches	63
Figure 41 : le processus de la 2eme étape.....	63
Figure 42 : Utilisation de la relation de transitivité.....	64
Figure 43 : le processus de la 3eme étape.....	64
Figure 44 : Retour des résultats	65
Figure 45 : le processus de la 4eme étape.....	65
Figure 46 : le résultat final	65
Figure 47 : architcture globalc	67
Figure 48 : les fichiers du package 'notre_package'.....	69
Figure 49 : teste de simulation pour une seule personne	70
Figure 50 : résultat du premier test.....	71
Figure 51 : test de simulation pour deux personnes.....	71
Figure 52: résultat du deuxième test.....	72
Figure 53 : test de simulation pour trois personnes	72

Figure 54 : résultat du troisième test.....	73
Figure 55 : test de simulation pour plus trois personnes	73
Figure 56 : résultat pour plus trois personnes	74
Figure 57 : test de l'algorithme dans un environnement réel	75
Figure 58 : le résultat du test	75
Figure 59 : Temps de réponse par rapport aux nombre de personnes	76

Liste des Tableaux :

Tableau 1 : les caractéristiques de la Kinect	35
Tableau 2 : montre la comparaison entre openni et SDK Windows	37

Résumé :

Cette thèse traite de la classification de personnes pour la robotique de services. Nous proposons un ensemble d'outils algorithmiques permettant de garantir, avant son déploiement, qu'une application, une fois installée sur une architecture robotique donnée, sera exécutée en temps réel. Nous nous plaçons dans un contexte temps réel strict avec des échéances de terminaison au plus tard. De plus, nous ne considérons que pour un robot guide muni d'une caméra RGB-D (Kinect), la perception de l'environnement doit prendre en compte les facteurs liés aux comportements sociaux habituels, à cette fin, des algorithmes efficaces pour la perception de personnes dans l'environnement du robot doivent être développés.

L'objectif de ce travail concerne le développement d'un algorithme qui permet à un robot guide d'exploiter les données, par rapport à un ou plusieurs attributs notamment le nombre, la direction, etc....

La technique effectuée a été implémentée sous forme de nœuds ROS puis testée et validée sur le robot mobile B21R du CDTA, Enfin nous expérimentons les conditions de faisabilité précédentes sur une plateforme réelle. Ces expérimentations montrent également que nos conditions de faisabilité s'avèrent opérationnelles pour le dimensionnement temps réel d'applications embarquées.

Mots clés : RGB-D, classification de personnes, robotique de service, Kinect, ROS, CDTA, B21R.

ملخص :

تتناول هذه الأطروحة استعمال الروبوتات من أجل تصنيف الأفراد, نقترح مجموعة من الخوارزميات و ذلك من أجل التأكد قبل تثبيت على الروبوت انه سيتم تنفيذ العمل بنجاح. لقد وُضعت تحت شروط اهمها الوقت الحالي حيث يشكل عامل رئيسي في عملنا , و ذلك راجع للروبوت المستعمل B21R المتمثل في كونه روبوت يؤدي مهام البشر.

الروبوت مزود بكاميرا من نوع كينكت تمتلك القدرة على التصوير العادي اضافة الى تصوير العمق.

في البيئة المحيطة بروبوت يجب ان نأخذ بعين الاعتبار السلوكيات الإجتماعية العادية للأفراد.

الهدف من هذا العمل هو تطبيق خوارزمية تكون بمثابة مرجع للروبوت, تسمح له بالاستغلال البيانات , مثل عدد الأفراد الخ.

التقنية التي تم تنفيذها هي من شكل تطبيقات ROS حيث تم ادراجها في الروبوت من أجل التأكد من صحتها.

في الأخير, التجارب المطبقة على الروبوت B21R من عند CDTA, تبين لنا ان العمل نجح بالفعل محترما جميع الشروط و التي من اهمها الوقت الحالي.

الكلمات الدلالية: تصنيف الأفراد , روبوت المهام , كينكت , ROS , CDTA , B21R.

Abstract:

This thesis is about classification of persons for service robotics.

We propose a set of algorithmic tools to ensure, before deployment, that the application once installed any robot's architecture it will be executed in real time.

We are working in a strict context in real time, with a dead line, in addition, we consider that for robot guide with a camera RGB-D the perception of the environment must consider the factors related to people social behaviors, by this effective algorithms must be developed.

The objective of this work concerns the development of an algorithm that allow a robot to exploit and use data, such as the number of persons ... etc.

Performed technic has been implemented in the form of ROS nodes and then tested and validated on the B21R of CDTA mobile robot, finally we are experimenting the previous feasibility conditions on a real platforme. These experiments also show that our conditions of feasibility are operational for real-time embedded applications.

Keywords: RGB-D, classification of persons, service robotics, Kinect, ROS, CDTA, and B21R.

Introduction générale

Depuis son avènement au milieu des années 70 et à nos jours, la robotique mobile continu de gagner en popularité dans les milieux universitaires et de recherche. Cette science demande beaucoup de créativité et de connaissance pluridisciplinaire, allant de la Mécanique à l'Electronique numérique et analogique, Programmation, Imagerie, Intelligence artificielle, Temps réel, etc. Cependant, ce domaine est en évolution constante, notamment en développant et améliorant les différentes plates-formes et outils et les capteurs existants utilisés dans robotique.

Pour être réellement autonome en environnement inconnu, un robot mobile doit en outre pouvoir percevoir et comprendre ce qui se passe autour de lui, en utilisant une variété intéressante de capteurs.

Les robots mobiles d'aujourd'hui disposent de différents capteurs, notamment ceux de type extéroceptifs, qui font que le robot puisse identifier un objet en temps réel.

Les capteurs visuels sont les plus appropriés pour ce dernier type de tâche. En effet, plusieurs applications pratiques de la vision par ordinateur ont été développées ces dernières années pour résoudre des problèmes cruciaux dans des applications comme la détection de l'être humain et des objets, le suivi, etc.

Pour un robot guide muni d'une caméra Kinect, la perception de l'environnement doit prendre en compte les facteurs liés aux comportements sociaux habituels, par exemple ne pas franchir un groupe ou un couple de personnes.

L'objectif de ce travail concerne le développement d'un algorithme qui permet à un robot guide d'exploiter les données issues d'une caméra RGB-D (Kinect) en fournissant une classification des personnes détectées, par apport à un ou plusieurs attributs.

La technique à proposer doit être implémenté sous forme de nœuds ROS (Robot operating System) puis testée et validée sur le robot B21R du CDTA.

Par la suite nous effectuons des tests en simulation puis en condition réel.

Ce mémoire s'articule principalement sur quatre chapitres :

- Le premier chapitre est consacré à la présentation de façon générale sur la robotique.

- Le deuxième chapitre présente les éléments de l'étude : détection et suivi, camera Kinect.
- Le troisième chapitre présente une recherche sur les différentes méthodes de classification existantes
- Le quatrième chapitre est dédié à la conception de l'approche dans laquelle nous détaillons chaque étape.
- Le dernier chapitre contient l'essentiel des résultats obtenus et leur interprétation.

Nous clôturons ce mémoire par une conclusion générale dans laquelle on récapitule ce qui a été fait, et mentionne des perspectives pour notre travail.

Chapitre 1

Introduction à la robotique

1. Introduction :

Le monde de la robotique est vaste, en partant de son histoire, il a fait un grand pas pour le milieu de la technologie moderne.

Depuis la création des automates jusqu'au terme robot, la technologie robotique a connu une grande évolution.

Dans ce premier chapitre, nous allons traiter l'histoire, les types, les classifications et le domaine d'application des robots.

Ensuite, nous allons évoquer la notion MIDDLEWARE robotique qui est le système d'exploitation du robot. Il existe plusieurs types de système, pour mieux comprendre nous allons en prendre un et l'expliquer en détail.

2. Définition et Historique :

2.1. Définition de la robotique :

Le terme « robot » découle de la langue slave et signifie textuellement « esclave ». Les machines automatiques sont appelées à remplacer les hommes dans les tâches les plus pénibles.

Une définition simple qui pourrait convenir aussi bien à un robot industriel qu'à un robot de service serait : un dispositif mécanique permettant de réaliser des tâches, en autonomie de décision pour une partie des actions élémentaires qui la composent. Le niveau d'autonomie doit être présent, mais ne porte pas nécessairement sur l'ensemble de la tâche. On peut la définir comme la capacité à réaliser ces actions élémentaires sur la base d'une consigne simple, après une prise d'information par le robot sur son environnement. Ainsi, la capacité à induire un comportement sur la base de ces informations et une interaction avec son environnement, et donc un certain niveau d'intelligence, est une caractéristique essentielle du robot. [1]

2.2. Histoire de la robotique :

On peut distinguer trois principales ères en robotique :

- les automates
- les robots n'étant pas dotés d'intelligence artificielle(IA)
- les robots disposant d'une intelligence artificielle

Les automates :

Un automate, contrairement à un robot (même s'il ne dispose pas d'intelligence artificielle), obéit uniquement à un programme préétabli, que ce soit de manière mécanique ou électronique. De ce fait, il n'y a aucune adaptation possible entre l'automate et son environnement. On attribue la paternité du tout premier automate de l'humanité à Architos de Tarente (IV^{ème} siècle avant J.C.).

Il s'agissait d'une représentation d'un pigeon capable de voler et étant propulsé par de la vapeur. Les toutes premières traces d'automates remontent à l'antiquité par Héron d'Alexandrie à l'1^{er} siècle après J.C., ses réalisations ornèrent les temples et les théâtres de la ville égyptienne.

En 1495 Léonard de Vinci présenta un chevalier humanoïde capable de s'asseoir, de relever sa visière et de bouger ses bras. L'automate le plus célèbre est le canard mécanique de Jacques De Vaucanson (Figure 1), capable d'ingurgiter de la nourriture et de la digérer. De Vaucanson aurait également présenté en 1738 un second automate représentant un homme jouant d'un instrument à vent. [2]

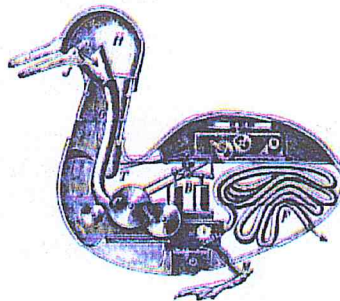


Figure 1 : Le canard mécanique de Jacques De Vaucanson

Les Robots sans IA :

Ce sont des robots qui disposent d'organes sensoriels, autrement dit des capteurs, pouvant influencer sur leurs comportements. Ils sont relativement adaptables à leur environnement.

Le chien électrique de Hammond et Meissner (1915) est le premier robot de ce genre. Il se déplace selon la luminosité de l'endroit grâce à son capteur optique.

Le chien Phillidog de Henri Piraux en 1928, et le renard de Ducrocq (1953) fonctionnent selon le même principe.

Walter Grey équipe en 1950 sa tortue cybernétique de capteurs tactiles et lumineux. (Figure 2).



Figure 2: Grey Walter et une tortue cybernétique sans carapace

Les Robots avec IA :

Un robot qui dispose d'une intelligence artificielle est un robot qui doit parvenir à effectuer une tâche par lui-même, sans aucune aide extérieure.

En 1973, l'université de Waseda présente le tout premier humanoïde "intelligent" dénommé Wabot-1(Figure 3). Il est doué de la vision, peut manipuler des objets, effectue un semblant de marche et est même capable de débiter une conversation en japonais.

Hi-T-Hand d'Hitachi en 1974 manipule des aiguilles à travers des trous grâce à la détection de force.

Vers la fin des années 70 Hans Morava présente les premiers robots capables d'évoluer à l'extérieur, de façon autonome.

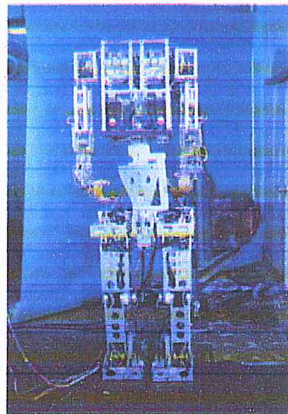


Figure 3 : Wabot-1

3. Classification Des Robots :

On peut Classifier les robots en 3 types :

3.1. Les manipulateurs :

- Les trajectoires sont non quelconques dans l'espace,
- Les positions sont discrètes avec 2 ou 3 valeurs par axe,
- La commande est séquentielle.

3.2. Les télémanipulateurs : appareils de manipulation à distance (pelle mécanique, pont roulant), apparus vers 1945 aux USA :

- Les trajectoires peuvent être quelconques dans l'espace,
- Les trajectoires sont définies de manière instantanée par l'opérateur, Généralement à partir d'un pupitre de commande (joystick).

3.3. Les robots :

- Les trajectoires peuvent être quelconques dans l'espace.
- L'exécution est automatique, Les informations. Extéroceptives peuvent modifier le comportement du robot. [3]

4. Types de Robots :

La Robotique est en constante évolution, on trouve beaucoup de domaines qui sont devenues indépendantes à la robotique, et cela est dû à la grande utilité à la contribution à l'élaboration de nombreux domaines scientifiques. Parmi ces domaines, nous trouvons :

4.1. Les robots Médical :

Ce sont des robots qu'on trouve dans les instituts médicaux, Ces robots offrent beaucoup d'aide dans le domaine de la médecine, où de nombreux hôpitaux sont devenus dépendants d'eux, par exemple en chirurgie. (Figure 4)

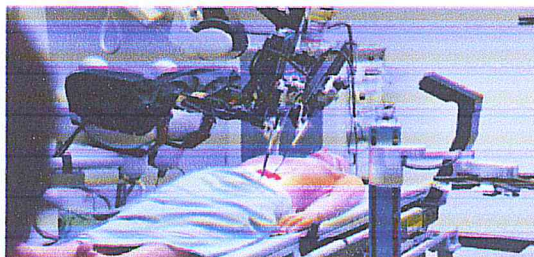


Figure 4 : un robot médical

4.2. Les robots divertissements :

Ce sont des robots utilisés pour le divertissement. Ceci est une catégorie très large. Il commence avec des robots jouets (figure 5) et se termine avec de vrais poids lourds tels que des bras de robot articulés utilisés comme simulateurs de mouvement.



Figure 5 : un robot de divertissement

4.3. Les robots Militaires :

C'est tous les Robots utilisés dans le domaine militaire. Ce type de robots comprend les robots d'élimination de bombe (Figure 6), différents robots de transport, drones de reconnaissance. Souvent, ces robots sont initialement créés à des fins militaires, mais plus tard, ils peuvent être utilisés dans l'application de la loi, de recherche et de sauvetage et d'autres domaines connexes.

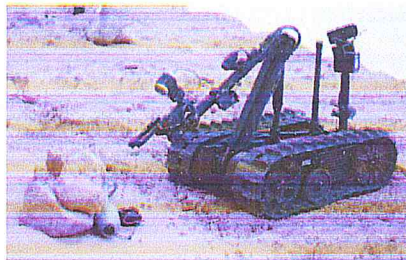


Figure 6 : Robot militaire

4.4. Les robots Domestique :

Ce sont des Robots utilisés à la maison, ce dernier comprend de nombreux appareils très différents, tels que les aspirateurs robotiques (Figure 7), les robots nettoyeurs de piscines, les balayeuses, les nettoyeurs de gouttière et d'autres robots qui peuvent faire différentes tâches. En outre, certains robots de surveillance et de télé présence peuvent être considérés comme des robots domestiques s'ils sont utilisés dans cet environnement.

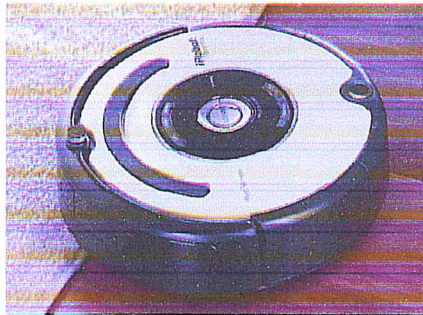


Figure 7 : Exemple Robots maison

4.5. Les robots Industriels :

Les robots industriels sont des robots utilisés dans un environnement de fabrication industrielle. Habituellement, ils sont articulés des bras (Figure 8) spécifiquement développés pour des applications telles que le soudage, la manutention, de peinture et d'autres. Si nous jugeons purement par l'application de ce type pourrait également inclure certains véhicules guidés automatisés et d'autres robots.

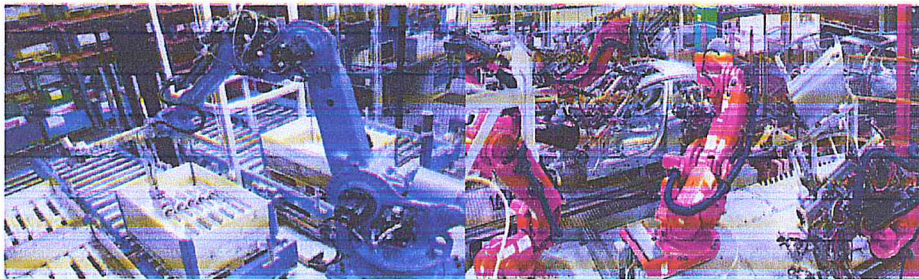


Figure 8 : Exemple Robots Industriels

4.6. Les robots de Services :

La robotique personnelle et de service ne constitue pas en tant que telle une chaîne de valeur dont la validité est générique. À l'instar d'Internet dans les années 1990, la robotique constitue aujourd'hui un cadre de développement d'outils et de produits qui répondront dans l'avenir à des demandes très variées, donc à des marchés dont les caractéristiques diffèrent fortement. Notamment dans sa partie commerciale, le modèle économique sur les différents marchés varie fortement suivant que l'on parle de marchés domestiques ou de la diffusion de robots d'assistance aux personnes en situation de handicap. Les contraintes d'exploitation, les coûts de fabrication, les niveaux de fiabilité.

[4]

Définition :

Un **robot de service** est un robot qui exécute des tâches utiles pour l'homme ou l'équipement à l'exclusion des applications d'automatisme industriel.

Catégories de la robotique de services :

Il existe deux catégories de robotique de service :

- a) **Robotique de service professionnelle.**
- b) **Robotique de service personnelle.**

a. Robotique de service professionnelle :

La robotique de service professionnelle intervient en assistance au travailleur dans un cadre professionnel. Ses fonctions sont principalement de soulager les professionnels de tâches répétitives ou dangereuses (dans une optique proche de la robotique industrielle), ou de les assister dans des interventions qui requièrent un niveau de précision ou des qualités inaccessibles à l'opérateur humain.

- **Robot agricole** (terrestre et drone, trayeuses...). John Deere développe avec iRobot et la Carnegie Mellon Université des engins autonomes pour l'agriculture et la défense. Les trayeuses automatiques sont classées par l'*International Federation of Robotics* dans la robotique de service, bien que, par ses fonctions et ses technologies, elles s'apparentent plus à la robotique industrielle. Sur le terrain agricole, hors trayeuses, les déploiements commerciaux sont aujourd'hui émergents.

- **Robot de nettoyage** (sol, verrière difficilement accessible). Ces robots constituent une part importante de la robotique de service professionnelle. Ces robots sont mis en œuvre notamment dans les espaces publics. Ils sont le fruit de développements spécifiques, et n'ont aujourd'hui que peu de liens – tant sur le plan des technologies que sur celui des acteurs – avec les robots de nettoyage domestiques.

- **Construction et démolition.** Les robots de construction sont utilisés de plus en plus pour des opérations spécifiques dans le bâtiment et le génie civil. Ils

sont en général associés à une technique particulière de construction (projection de béton par exemple). Ce sont dans tous les cas des machines spéciales développées pour une application particulière. Les principaux fabricants et concepteurs de robots de construction sont également des professionnels du BTP et du génie civil (par exemple Tasei Corporation ou Fujita Corporation) ; les missions possibles vont de la construction à la reconstruction en passant par le déblaiement dans des opérations de secours. La démolition en tant que telle est surtout robotisée sur le démantèlement d'installations dangereuses (nucléaires notamment).

- **Robot logistique** (véhicule de transport automatique de personnes et de biens, tortues dans les hôpitaux...). Les robots logistiques sont également une voie de développement intéressante, notamment en raison de la possibilité de développer des machines produites en série : la logistique est un secteur vaste, qui met en œuvre des procédures et équipements pour partie standardisés. Notons que l'on inclut dans la logistique des robots de tri et de préparation de colis qui n'entrent pas pleinement dans la définition que nous avons donnée de la robotique de service.

- **Robot de relations publiques** (accueil, guide, transport de personne en visite interactive...). Les robots d'accueil ou d'assistance dans les lieux publics sont aujourd'hui l'objet d'expérimentations plus que d'un réel développement commercial à l'échelle d'une filière. Les robots de relations publiques sont mis en œuvre dans des lieux tels que des musées ou des centres commerciaux, pour aider les visiteurs à s'orienter, leur fournir des renseignements, voire apporter un aspect ludique à la muséographie du lieu.

- **Robot médical** (assistance à l'intervention chirurgicale, intervention/examen à distance, assistance à la personne en manque d'autonomie dans sa rééducation et/ou la vie courante...). Le domaine médical est également un marché important pour le développement de la robotique de service professionnelle. Il s'agit là d'un marché ouvert aux innovations, en forte demande et avec des attentes précises des professionnels de santé et des patients. Les axes de développement visent à l'assistance des médecins (robots d'assistance chirurgicale par exemple), du personnel paramédical (assistance à la manipulation des personnes alitées, fauteuils roulants robotisés), des patients (aide à la rééducation, prothèses ou orthèses robotisées – dont exosquelettes à terme). Ces dernières catégories de systèmes robotisés (fauteuils robotisés, orthèse,

exosquelette...) étant plus à classer dans les robots de services personnels du fait de leur utilisation directe par le patient dans sa vie courante.

- **Robot de surveillance et de sécurité sur site industriel ou tertiaire** (terrestre, drone, maritime). Ces robots constituent clairement l'un des trois segments qui seront étudiés dans le cadre du volet 2 de ces travaux. Ils reposent sur deux types d'actions : d'une part la surveillance au sens de gardiennage, c'est-à-dire la prévention des intrusions physiques ; d'autre part la surveillance au sens de *monitoring*, qui assiste l'opérateur humain dans la surveillance environnementale ou industrielle des sites. Le premier champ d'applications est aujourd'hui le plus développé. Un exemple de produit sur ce segment est le Jazz Security (Figure 9) de Gostai, vendu près de 10 000 euros.



Figure 9 : Robot d'inspection de surveillance Jazz Security

- **Robot d'assistance au geste dans le milieu du travail.** Également appelés cobots (Figure 10) pour robots collaboratifs, il s'agit de systèmes robotiques assistant le geste du professionnel en environnement de travail (atelier, hangar de stockage...). Le principe de leur fonctionnement est d'assister le geste de la personne en apportant une force complémentaire (diminution des efforts, une précision accrue, etc.).



Figure 10 : Robot d'assistance au geste (ou cobots) RB3D

Au-delà des applications décrites ci-dessus, il n'est pas possible d'établir une liste exhaustive des applications potentielles. Des fournisseurs de technologies développent les briques nécessaires à la conception de robots, que les intégrateurs et bureaux d'études s'approprient au cas par cas pour des usages répondant à une demande ponctuelle dans diverses utilisations. [5]

b. Robotique de service personnelle :

La robotique personnelle constitue un champ potentiellement aussi foisonnant que la robotique de service professionnelle. Les utilisations sont possibles pour répondre à un grand nombre de besoins et d'attentes. À travers la description suivante, on tente d'en dresser un inventaire fourni.

- **Robot domestique** (réalisant des tâches domestiques d'entretien de la vie courante) : il constitue aujourd'hui le principal marché domestique pour des robots évolués. Deux fonctions sont aujourd'hui des réalités commerciales : l'aspirateur d'une part, avec des ventes croissant exponentiellement dans le monde, et un panel de fournisseurs relativement important, même si l'un d'entre eux, iRobot (US), semble dominer le marché ; d'autre part les robots de lavage des sols, qui sont aujourd'hui émergents sur le marché avec peu d'acteurs ayant mis un produit sur le marché (iRobot Scooba et Royal Appliance International GmbH – marque DirtEvil – par exemple). Le marché est ouvert par les aspirateurs, permettant d'envisager le développement de nouvelles applications en fonction des développements techniques et des choix marketing des fabricants. Ainsi devraient apparaître dans les prochaines années des robots laveurs de vitres, des robots majordomes ou des assistants personnels ...

• **Robot thérapeutique.** Derrière cette appellation se groupent un grand nombre de fonctions possibles. Dans le cadre de ces travaux, on y placera les robots qui permettent de simuler une présence. Ces robots « affectifs » (Figure 11) trouvent aujourd'hui de réels débouchés, notamment au Japon. On citera pour l'exemple le robot Paro, en forme de bébé phoque, qui trouve sa place chez les personnes âgées au Japon, et auquel on prête des vertus dans le cadre des syndromes d'Alzheimer. Au-delà de cette réussite commerciale annoncée, le robot Kaspar fait l'objet de travaux de recherche dans le cadre de l'assistance à l'apprentissage des relations sociales chez des enfants autistes. Des robots comme Paro trouvent leur place chez le patient et sont donc groupés dans les robots personnels. On voit cependant sur ce cas précis la difficulté à établir des frontières entre les grandes catégories de robots de service.

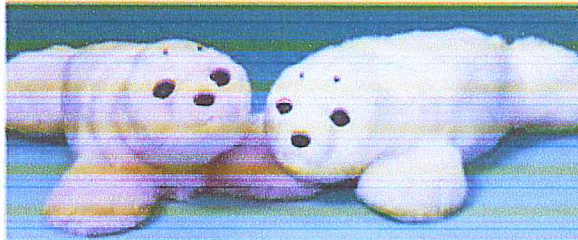


Figure 11 : Robot thérapeutique Paro

• **Robot d'assistance aux personnes en perte d'autonomie** (personnes handicapées, personnes âgées, malades). Ce secteur sera traité spécifiquement dans le volet 2 des travaux. Cette catégorie est connexe à la précédente, bien qu'assistant les personnes sur un autre plan. La perte d'autonomie est aujourd'hui un problème humain et économique : humain à travers la dégradation des conditions de vie des personnes, économique *via* le coût de l'accueil des personnes dans des structures spécialisées. Elle a pour origine des causes extrêmement diverses, et peut-être aussi bien momentanée que permanente. La conséquence en est que les applications d'assistance couvrent potentiellement des fonctions très larges, allant du système d'alerte ou de télé présence pour des personnes valides mais présentant un risque d'accident, à des systèmes d'assistance physique pour des opérations de la vie courante ou les déplacements. En anticipant sur le volet 2, on peut d'ores et déjà présumer que les systèmes de télé présence et de télé diagnostic sont une voie de développement commercial de ces applications à débouchés relativement rapides : les premières offres commerciales existent, mais ne sont pas intégrées dans les

prestations d'alerte et d'assistance à domicile. Les systèmes robotisés d'assistance physique pour les personnes dépendantes représentent également un challenge important de développement pour l'amélioration de la vie courante des personnes handicapées mais le développement commercial de ses systèmes est plus difficile du fait des petites séries produites.

- **Robot éducatif.** Le robot éducatif se conçoit selon deux approches distinctes. D'une part dans une fonction d'assistance à l'éducation des enfants, en créant une réelle interaction pour des activités à caractère pédagogique. C'est par exemple le cas du robot iRobiQ de Yujin, qui équipe les jardins d'enfant en Corée. D'autre part, comme support à l'éducation à la robotique. Dans ce cas, on parle plutôt soit de kits de robotique, soit de robots programmables. Les robots Lego Mindstorm constituent un support pédagogique courant.

- **Robot de jeux ou robots jouets.** Les robots ludiques présentent des caractéristiques très variées, dépendant de la gamme du produit : depuis le robot-peluche très simple dans ses fonctionnalités, jusque – à l'autre extrême – l'Aibo de Sony (aujourd'hui plus commercialisé), entièrement programmable, en passant par les Lego Mindstorm (robots à construire). [6]

En résumé, les critères distinctifs entre les robots professionnels et personnels peuvent être :

- **L'usage du robot** : le robot personnel exécute des tâches rendant un service à la personne particulière dans son cadre de vie (domicile, espaces publics, hôpital, etc.). *A contrario*, le robot professionnel rend un service à la personne dans l'accomplissement des tâches de son métier.
- **Le critère d'accessibilité économique** : un robot personnel est un robot « abordable » économiquement pour les particuliers. Ceci n'implique pas nécessairement que le prix du robot lui-même soit bas, mais que le mode de financement de son utilisation (soutien financier au titre de l'assurance maladie, location, abonnement à un service, etc.) le rende accessible.
- **Une analyse de rentabilité d'exploitation et d'utilisation distinctes** (dissemblable sur un achat professionnel et un achat grand public) : un calcul purement économique pour le robot professionnel, éventuellement basé sur ces critères de confort ou de divertissement pour le robot personnel.

- **Des modes d'interaction différents** : le domaine professionnel se caractérise notamment par une capacité à déployer des systèmes nécessitant une formation de la part des opérateurs. Ceci est beaucoup plus difficile à accepter du point de vue du consommateur.
- **Un aspect supervision d'ensemble et une association ou non avec un service distant pour remplir une part de ses tâches** : le robot professionnel est souvent partie prenante d'un système plus large
- Éventuellement, une quantité de fabrication en série très différente, impliquant des choix distincts de conception et fabrication.

De ce fait, si certains acteurs de la filière trouveront des marchés sur les deux créneaux, les différences sont suffisantes pour justifier que des entreprises se spécialisent sur l'un ou l'autre, les modèles économiques étant tout à fait différents (en règle générale).

- **Robot de surveillance domestique.** Les robots domestiques de surveillance sont envisagés pour couvrir plusieurs types d'applications : la première d'entre elles, en termes de temps d'accès au marché, est certainement la surveillance anti-intrusion. Les prémices de tels robots sont présentes par exemple sur les derniers-nés des aspirateurs LG, qui donnent accès à travers un *smartphone* aux images acquises par le système vidéo de l'appareil. [7]

LES DOMAINES APPLICATIFS :

Pour cadrer au mieux le périmètre de la robotique personnelle et de service, telle que nous l'entendons dans la présente étude, il nous paraît important de l'illustrer par l'exemple.

A. Les robots d'intervention :

La robotique d'intervention (traduction de *Field robotics*) n'est pas – globalement – au cœur de nos travaux ; elle est cependant souvent incluse dans la robotique de service dans la littérature.

Les robots d'intervention sont en général des robots télé opérés (opérés à distance) par des commandes directes (joysticks, bras maître et autres organes de commande physiques ou virtuels), ou en semi-autonome par des ordres de haut niveau pour réaliser et enchaîner des tâches (« va à telle position », « prends cette pièce désignée sur un écran »)... L'opérateur est donc systématiquement dans la

boucle de commande du robot afin d'interagir et l'aider à réaliser sa mission en fonction des évolutions de l'environnement matériel et humain.

Les robots d'intervention sont en général utilisés pour remplir des tâches dans des environnements difficiles d'accès ou dangereux pour les humains, ou encore lorsque l'absence d'humain rend l'exploitation plus aisée ou plus efficace.

B. Les robots de Défense :

Théâtre d'opérations (reconnaissance, déminage (Figure 12), surveillance de zone, logistique...) avec des robots terrestres et des drones aériens. Les robots militaires entrent pour la plupart dans le champ des robots d'intervention. Ils portent des fonctions de reconnaissance, de surveillance, de déminage ou de destruction.

La décision de tir sur les robots armés est pour l'instant, pour des raisons d'éthique, sujette à la décision de l'opérateur humain.



Figure 12 : Robot de déminage Packbot 510 EOD

C. Sécurité civile :

La robotique de sécurité civile est mise en œuvre notamment lors des interventions sur catastrophes naturelles. Des robots d'exploration, notamment, permettent d'explorer des lieux inaccessibles à la recherche de victimes. Dans ce champ, on trouve également les robots utilisés par les forces de l'ordre pour le désamorçage ou la destruction des colis piégés ; dans ce dernier cas, la qualification de « robot » pourrait cependant être discutée à la lumière de la définition que nous proposons.

- Nucléaire :

Le nucléaire est un domaine précoce de développement de la robotique d'intervention. Notamment, il semble que l'expertise initiale du CEA sur la robotique se soit construite sur ce type d'application. Elle concerne essentiellement l'intervention en environnement irradié, nocif pour l'opérateur humain. Elle a

donné lieu au développement de technologies « durcies » pour résister à des niveaux plus ou moins élevés de rayonnement.

D. Sous-marin :

La robotique sous-marine est également un champ de développement important pour des applications militaires (surveillance sous-marine), pétrolières (exploration, exploitation), de recherche (exemple de la recherche des boîtes noires d'avions accidentés), etc. Les robots sous-marins, comme pour l'exploration spatiale, ont l'avantage de ne pas nécessiter l'emport d'opérateurs humains, générateurs de contraintes d'exploitation coûteuses (espace vital, air, maintien de la pression, etc.).

- Inspection et maintenance :

L'inspection en milieux spécifiques s'intéresse à l'intervention à des endroits où l'homme ne peut matériellement pas intervenir à un coût acceptable. Notamment, la réparation de fuites sur les canalisations est une application relativement courante de la robotique, qui évite des travaux de génie civil pour une intervention humaine

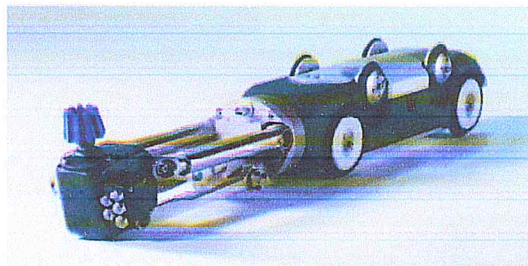


Figure 13 : Robot d'inspection et réparation de canalisation KASRO 4.0 DN200-DN600

E. Exploration spatiale :

Les robots sont aujourd'hui le moyen privilégié de l'exploration spatiale. Les missions vers Mars en sont un exemple significatif : les délais de transmissions de l'information n'autorisent pas une télé opération en temps réel par un manipulateur distant (sur Terre). Ce type d'applications est porteur d'innovations importantes en termes de perception, de locomotion, de robustesse, etc. [8]

4.7. Périmètre D'étude :

Ainsi, sans nier les liens entre les différents pans de la robotique, voire la filiation technologique qui existe entre eux, le cœur de la présente étude est la robotique personnelle et de service et les écosystèmes qui se développent autour de ces domaines d'application de la robotique.

Ainsi, le cadre de l'étude est la robotique personnelle et de service. Ce cadre demande à être précisé avant d'aller plus avant dans l'analyse, mais on peut d'ores et déjà tenter de définir le périmètre de l'étude par l'exemple, en classant les applications selon trois grandes catégories :

- Les applications cœur de cible, qui sont de manière évidente et consensuelle classées dans cette catégorie d'outils :
 - Robot de surveillance et de sécurité sur site industriel
 - Robot domestique (nettoyage, surveillance...)
 - Robot éducatif (domestique)
 - Robot compagnon (domotique, éducatif, ludique...)
 - Robot d'assistance aux personnes en perte d'autonomie (tous milieux)
- Les applications qui ressortent de la robotique personnelle et de service, mais ne sont pas au cœur de l'étude. Ces applications sont cependant pleinement prises en compte dans le présent volet des travaux sur le benchmark :
 - Robot de relations publiques
 - Robot jouet et ludique
 - Robot éducatif (en milieu scolaire)
- Les applications hors périmètre, mais qui intéressent l'objet de l'étude par les relations qui peuvent exister – *via* des acteurs ou des technologies – avec les applications cœur de cible :
 - Robot agricole
 - Robot de nettoyage industriel
 - Construction/démolition
 - Robot logistique et véhicules automatiques
 - Robot d'assistance au chirurgien

Notamment, on souligne que l'automobile, bien qu'utilisatrice de plus en plus de technologies robotiques, n'entre pas de plein champ dans le sujet de ces travaux. De la même manière que pour la robotique militaire par exemple, il est pourtant exclu de les oublier dans l'analyse : la robotique est et restera un secteur défini par une certaine vision de l'autonomie et du mode de fonctionnement des systèmes (largement alimenté dans l'imaginaire collectif par la littérature et le cinéma de science-fiction, des technologies constitutives, mais non exclusives, enfin et surtout par un développement à travers des marchés d'opportunité. Il est donc difficile) voire impossible – de fixer des frontières nettes à l'objet de nos travaux. [9]

5. Middleware robot :

Dans cette section, on explore les exigences des technologies robot middleware en général et on va examiner certaines approches de robots existants tels que YARP, OpenRDK, OpenRTM-AIST et ROS.

5.1. Définition :

Middleware robotique est un middleware de logiciel qui étend la communication middleware tels que CORBA ou ICE. Il fournit des outils, des bibliothèques, des API et des lignes directrices pour soutenir la création et le fonctionnement des deux composants du robot et des systèmes de robots.

Robot middleware agit également comme une colle qui établit une connexion entre les parties de robot de façon transparente.

5.2. Existant robot de middleware :

Il existe plusieurs middlewares pour la robotique, citons :

- YARP
- OpenRTM
- OpenRDK
- ROS

Dans Notre Project le middleware ROS (Robot Operating System) a été imposé par notre établissement de stage le CDTA, de ce qu'on va détailler juste-après.

5.3. ROS :

Désormais, la robotique est un domaine très actif qui fait l'objet de plusieurs sujets de recherche scientifique qui ont un impact sur notre communauté (socioéconomique, aides humanitaires, médecine... etc.)

De nos jours, la recherche robotique est beaucoup plus simple qu'avant, et cela grâce au développement et à l'évolution des outils utilisés notamment les systèmes d'exploitation et les plats-formes robotique soft et hard.

Dans cette partie du chapitre nous présentons l'un de ces outils qui est très sollicité à travers le monde, vu le nombre de fonctionnalités et services offres, il s'agit bien du système exploitation Robot ROS (Robot Operating System).

Définition ROS :

Comme son nom l'indique (Robot Operating System) est un système d'exploitation robotique qui s'occupe entre autre de gérer le matériel et de fournir des services pour l'exécution de diverses application soft, cependant cette appellation convient beaucoup plus à la première version (qui est un système d'exploitation proprement dit) qu'à la version de ROS qui est connecter comme un middleware (une plate-forme ou logiciel qui sert à soutenir et connecter les application soft en créant des réseaux de communication).

Cet outil ROS est sous licence libre (open source), et offre plusieurs fonctionnalités à différents niveaux et peut être exécuté sur plusieurs plate-forme robotique, il assure notamment une communication entre les différents modules robotique à travers un mécanisme très riche et très organisé (qu'on détaillera par la suite) ; ROS est monté principalement sur le système d'exploitation Linux.

Apport de l'outil ROS :

Avant l'arrivée des outils robotique, chaque concepteur de robot, chaque chercheur en robotique passait un temps non négligeable à concevoir matériellement son robot ainsi que le logiciel embarqué associé (notamment le soft qui le contrôle). Cela demandait des compétences dans plusieurs disciplines, notamment en mécanique, électronique et programmation embarquée. Généralement, les programmes ainsi conçus correspondaient plus à de la programmation embarquée (programmation sur les plateformes robotique), proche de l'électronique, qu'à de la robotique proprement dite, telle que nous pouvons la

rencontrer aujourd'hui dans la robotique de service. La réutilisation des programmes était non triviale, car ils étaient fortement liés au matériel sous-jacent. [29]

Comme la robotique nécessite des compétences très différentes, qui sont généralement hors de la portée d'une seule et même personne (d'où la notion d'équipe dans la recherche), ROS présente un autre avantage, celui de rassembler des savoir-faire de différentes disciplines. [30]

En effet, concevoir et programmer un robot, c'est :

- Gérer le matériel en écrivant les pilotes.
- Gérer la mémoire et les processus.
- Gérer la concurrence et la fusion de données.
- Proposer des algorithmes de raisonnement abstrait faisant largement appel à l'intelligence artificielle.

Principe de conception de ROS : [32]

La philosophie de conception de ROS se résume dans les principes suivants :

- Egale à Egale : (Peer-to-Peer) c'est une architecture couplée à un système de tampon (buffering), elle permet à chacun des acteurs de l'architecteur software et hardware du robot de dialoguer en direct avec un autre acteur, de manière synchrone ou asynchrone en fonction des besoins.
- Multi langage : ROS est neutre d'un point de vue langage et peut être programmé en différents langages. La spécification de ROS intervient au niveau message. Les connections peer to peer sont négociées en XML-RPC qui existe dans un grand nombre de langages. Pour supporter un nouveau langage, soit on redéfinit les classes C++ (ce qui a été fait pour le client Octave par exemple), soit on écrit les classes permettant de générer les messages. Ces messages sont décrits en IDL (interface définition langage).
- Basé sur des outils : plutôt qu'une exécution (runtime) monolithique, ROS a adopté un design microkernel qui utilise un grand nombre de petits outils pour compiler et exécuter les différents programmes. Lorsqu'on parcourt les tutoriaux ROS, on apprend à nous servir de plusieurs commandes permettant de manipuler les différents niveaux du « système de fichier ROS » notamment les nœuds. Chaque

commande est en fait, un exécutable. L'avantage de cette solution est qu'un problème sur un exécutable n'affecte pas les autres, ce qui rend le système plus robuste et plus évolutif qu'un système basé sur une exécution centralisée.

Ajouté à ces trois principes, la conception de ROS est faite sous licence libre (open source), ce qui veut dire qu'on peut le modifier et le customiser selon nos besoins.

Système de Fichiers ROS :

Le système de fichier est organisé dans une structure hiérarchique basé sur six principaux points :

- Le package ou Paquet logiciel : ROS est basé sur la notion de paquets, qui communiquent entre eux grâce à ce qu'on appelle « un master » ; le package est donc l'unité principale d'organisation logicielle de ROS.
- Manifest: c'est un fichier qui fournit des informations concernant les packages, il définit les dépendances et les bibliothèques utilisées par le compilateur.
- La pile (stack) : est une collection de packages, elle contient un ensemble de fonctionnalités (telles que la navigation et la localisation). Elle est sous forme d'un répertoire (dossier) qui contient d'autres répertoires (chaque package dans cette pile, est contenu dans un répertoire), ainsi qu'un fichier de configuration nommé stack.xml (c'est le Manifest de la pile).
- Manifest de la pile : c'est un fichier qui fournit des informations concernant la pile (on y trouve les différents packages contenus dans la pile et d'autres informations).
- Type Message : définit la structure de données pour les messages.
- Type Service : définit la structure de données respectives de la demande (requête) et celle de la réponse d'un service.

Système de traitement et de calcul ROS :¹

Le principe de base d'un OS robotique est de faire fonctionner en parallèle un grand nombre d'exécutables qui doivent pouvoir échanger de l'information de manière synchrone ou asynchrone. Par exemple, un OS robotique doit interroger à une fréquence définie les capteurs du robot (capteur de distance à ultrasons ou infrarouge, capteur de pression, capteur de température, gyroscope, accéléromètre, caméras, microphones...), récupérer ces informations, les traiter (faire ce que l'on appelle la fusion de données), les passer à des algorithmes de traitement (traitement de la parole, vision artificielle, localisation et cartographie simultanée,) et enfin contrôler les moteurs en retour. Tout ce processus s'effectue en continu et en parallèle. D'autre part, l'OS robotique doit assurer la gestion de la concurrence afin d'assurer l'accès efficace aux ressources du robot.

Nous décrivons ci-dessous les concepts regroupés dans ROS sous le nom de « ROS Computation Graph » et qui permettent d'atteindre ces objectifs. Il s'agit des concepts utilisés par le système en cours de fonctionnement tandis que le « ROS FileSystem » décrit dans le paragraphe précédent correspond aux concepts statiques.

- **LES NOEUDS** : ROS répond à tout cette problématique grâce à des notions de base simples. La première notion est la notion de nœud. Dans ROS, un nœud est une instance d'un exécutable. Un nœud peut correspondre à un capteur, un moteur, un algorithme de traitement, de surveillance... Chaque nœud qui se lance se déclare au Master. On retrouve ici l'architecture microkernel où chaque ressource est un nœud indépendant.
- **LE MASTER** : Le Master est un service de déclaration et d'enregistrement des nœuds qui permet ainsi à des nœuds de se connaître et d'échanger de l'information. Le Master est implémenté via XMLRPC. Le Master comprend une sous-partie très utilisée qui est le Parameter Server. Celui-ci, également implémenté sous forme de XMLRPC, comme son nom l'indique est une sorte de base de données centralisée dans laquelle les nœuds peuvent stocker de l'information et ainsi partager des paramètres globaux.
- **LES TOPICS** : L'échange de l'information s'effectue soit de manière asynchrone via un topic ou de manière synchrone via un service. Un topic est un système de transport de l'information basé sur le système de l'abonnement / publication (subscribe / publish).

¹ <http://www.generationrobots.com/fr/content/55-ros-robot-operating-system>

Un ou plusieurs nœuds pourront publier de l'information sur un topic et un ou plusieurs nœuds pourront lire l'information sur ce topic. Le topic est en quelque sorte un bus d'information asynchrone un peu comme un flux RSS. Cette notion de bus many-to-many asynchrone est essentielle dans le cas d'un système distribué. Le topic est typé, c'est-à-dire que le type d'information qui est publiée (le message) est toujours structuré de la même manière. Les nœuds envoient ou reçoivent des messages sur des topics.

- **LES MESSAGES** : Un message est une structure de donnée composite. Un message est composé d'une combinaison de types primitifs (chaines de caractères, booléens, entiers, flottants...) et de message (le message est une structure récursive). Par exemple un nœud représentant un servomoteur du robot, publiera certainement son état sur un topic (selon ce que vous aurez programmé) avec un message contenant par exemple un entier représentant la position du moteur, un flottant représentant sa température, un autre flottant représentant sa vitesse La description des messages est stockée dans *nom_package/msg/monMessageType.msg*. Ce fichier décrit la structure des messages.
- **LES SERVICES** : Le topic est un mode de communication asynchrone permettant une communication many-to-many. Le service en revanche répond à une autre nécessité, celle d'une communication synchrone entre deux nœuds. Cette notion se rapproche de la notion d'appel de procédure distante (remote procedure call). La description des services est stockée dans *nom_package/srv/monServiceType.srv*. Ce fichier décrit les structures de données des requêtes et des réponses.

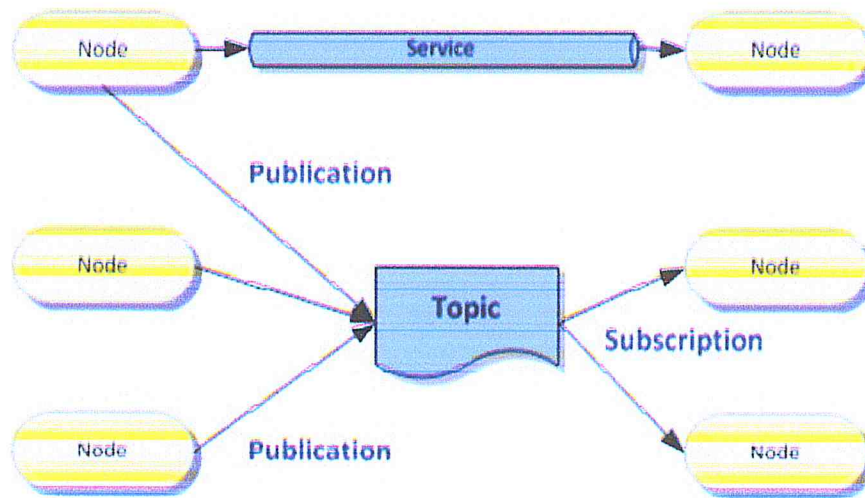


Figure 14 : fonctionnement du système ROS

6. Conclusion :

Pour conclure la robotique est un domaine très complexe dans le sens technique ou artistique, nous l'avons constaté par la diversité des types de robot soit personnels ou professionnels.

Dans ce chapitre nous avons donné une vue panoramique sur le monde de la robotique ancienne et moderne.



Chapitre 2

Détection et suivi

1. Introduction :

Les vidéos sont en fait des séquences d'images dont chacune est appelée Frame, affichée en fréquence assez rapide pour que l'œil humaine puisse percevoir la continuité de son contenu, il est évident que toutes les techniques de traitement d'image peuvent être appliquées à des images individuelles : En outre, le contenu de deux cadres consécutifs est généralement très proche.

Le Contenu visuel peut être modélisé comme une hiérarchie d'abstractions, au premier niveau sont les pixels bruts avec les informations de couleur ou de luminosité. Une couche d'abstraction plus élevée peut combiner et interpréter ces caractéristiques comme les objets et leurs attributs.

Le niveau le plus élevée représente les concepts humains de niveau impliquant un ou plusieurs objets et relations entre eux.

La détection d'objets dans des vidéos consiste à vérifier la présence d'un objet dans des séquences d'images et éventuellement la localisation précisément pour la reconnaissance. Le suivi d'un objet consiste à surveiller un objet et les changements temporelle lors d'une séquence vidéo y compris sa présence, position, taille, forme, etc. cela en résolvant le problème correspondance temporels, le problème consiste à faire correspondre la région cible en images successives d'une séquence d'image prises à des intervalles de temps rapprochés. Ces deux processus sont étroitement liés car le suivi habituellement commence par détecter des objets tout en détectant le même objet à plusieurs reprises dans la séquence d'images ultérieures est nécessaire d'aider et de vérifier le suivi.

2. La Détection d'objet et les approches de suivi :

2.1. Détection Orientée objet :

Dans la détection de l'orienté objet, normalisation des caractéristiques de l'image et l'enregistrement (alignement) des points de référence sont importants. Les images peuvent doivent être transformés en un autre espace pour gérer les changements dans l'éclairage, la taille et l'orientation. Une ou plusieurs caractéristiques sont extraites et les objets d'intérêt sont modélés en fonction de ces caractéristiques.

2.2. Approches base sur la forme :

La détection d'objet basée sur la forme est un des problèmes les plus difficiles en raison des difficultés de la segmentation dans les images. Afin de détecter et déterminer la frontière d'un objet, une image doit être prétraitée. L'algorithme de prétraitement, ou filtre dépend de l'application. Différents types d'objets tels que des personnes, des fleurs et des avions peuvent exiger différents algorithmes.

Lorsque l'objet est détecté et situé, on y applique des algorithmes de détection des contours et de suivi. La caractérisation de détection et de la forme des objets devient plus difficile pour les scènes complexes où il y a beaucoup d'objets avec les occlusions et les ombres. [14]

2.3. Approches base sur la couleur :

Contrairement aux approches basées sur la forme la couleur est relativement constante dans les changements de point de vue et elle est facile à acquérir. Bien que la couleur ne soit pas toujours appropriée comme unique moyen de détection et de suivi d'objets, son exploitation reste souhaitable grâce au faible coût de calcul de ses algorithmes.

[10] mis au point un algorithme permettant de détecter et de localiser les véhicules ou piétons en temps réel à l'aide technique basé sur l'histogramme de couleur, en construisant un modèle de mélange gaussien pour décrire la répartition de la couleur dans la séquence d'images et de segmenter l'image en arrière-plan et objets. Occlusion de l'objet ont été conduite à l'aide d'un tampon d'occlusion.

[11] traite le suivi de multiples visages en temps réel en utilisant des repères de couleur. Cette méthode de suivi simple repose sur le suivi des régions de la même couleur normalisée d'une image à une autre. Ces régions sont définies dans l'étendue de l'objet à suivre avec une taille fixe et des positions relatives. Chaque région est caractérisée par un vecteur de couleur calculé par échantillonnage sous les pixels dans la région, qui représente la couleur moyenne des pixels dans cette région.

2.4. Détection basé sur la correspondance de modèle :

Si un modèle décrivant un objet spécifique est disponible, la détection d'objet revient à mettre en correspondance les caractéristiques entre le modèle et la séquence d'images sous analyse.

La Détection d'objet avec une correspondance exacte est généralement coûteuse et dépend de la qualité des détails du degré de précision fournie par le modèle objet.

Il y a deux types de correspondance de modèle d'objet :

- Fixe
- Déformable

Fixe :

Les modèles fixes sont utiles lorsque les formes de l'objet ne changent pas par rapport à l'angle de vision de la caméra. Deux techniques principales ont été utilisées :

a. Soustraction d'image :

Dans cette technique, la position du modèle est déterminée en minimisant la fonction de la distance entre le modèle et les diverses fonctions dans l'image. Bien que les techniques de soustraction d'images nécessitent moins de temps de calcul que les techniques de corrélation. Elles s'effectuent bien dans des environnements restreints où des conditions telles que l'intensité de l'image et l'affichage des angles entre le modèle et les images contenant ce modèle sont les mêmes.

b. Corrélation :

Cette technique est généralement insensible aux effets de bruit et d'éclairage dans les images [12], mais souffre d'une grande complexité computationnelle causée par la sommation sur l'ensemble du modèle. Le point de corrélation peut réduire la complexité du calcul à un petit ensemble de points choisis avec soin pour les sommations.

Déformable :

Les approches de correspondance de modèle déformable sont plus appropriées pour les cas où les objets varient en raison des déformations rigides et non rigides [13]. Ces variations peuvent être causées soit par la déformation de l'objet en soi. En raison de la nature déformable d'objets dans la plupart des vidéos, des modèles déformables sont plus intéressants dans le suivi des tâches.

Dans cette approche, un modèle est représenté sous forme de bitmap décrivant les contour/bords caractéristiques d'une forme de l'objet. Une transformation probabiliste sur le contour du prototype sert à déformer le modèle pour s'adapter à des bords saillants dans l'image d'entrée.

3. Détection de Mouvement :

Détecter le déplacement d'objets, ou détection de mouvement, évidemment a une signification très importante dans la découverte de l'objet vidéo en mouvement.

La détection du mouvement complique le problème de détection d'objet en ajoutant la notion du temps aux exigences de changement. Il fournit également une autre source d'information pour la détection et le suivi.

Une grande variété d'algorithmes de détection de mouvement a été proposée.

Citons :

- Détection de mouvement par méthode de segmentation.
- Détection de mouvement par méthode de binarisation.
- La Détection de mouvement par le changement d'images.

3.1. Détection de mouvement par segmentation :

La segmentation d'images est utilisée pour séparer une image ou groupe d'images de telle façon que chacune des régions ait une caractéristique qui la distingue des autres, plusieurs techniques ont été proposées mais elles ont toutes un même objectif c'est l'infériorisation (minimisation) d'une fonctionnelle de distance, ce qui donne un résultat qui est sensiblement le même quel que soit l'algorithme utilisé.

3.2. Détection de mouvement par méthode de binarisation :

La binarisation est une technique qui produit deux classes de pixels, la détection de mouvement en utilisant cette technique est basée sur la détection du changement temporel au niveau pixels ou groupe de pixels, en utilisant une valeur seuil prédéfini pour déterminer la classification mouvement/non mouvement.

Remarque : il Ya une grande différence entre les deux technique :

- La binarisation comme son nom indique (binaire) elle produit deux classes de pixels.
- La segmentation au contraire elle produit plusieurs classes de pixels.

Détection par changement d'image :

- Approches base sur la région avec une image référence :

Cette méthode est appliquée généralement lorsqu'on possède une caméra fixe. Elle se base sur la comparaison des éléments fixe dans une image appelée référence et les éléments dans les autres images, pour déterminer les éléments qui sont en mouvement.

Donc pour déterminer les éléments en mouvement il suffit de faire la différence entre les deux images.

Cette méthode est très utilisée dans la détection du mouvement d'objet mais elle à une difficulté, c'est comment construire une image référence. Une solution simple consiste à mettre la camera dans un environnement où aucun objet n'apparait. Ensuite, il faudrait sélectionner l'image manuellement, mais ce n'est pas toujours le cas. Dans [15] Makarov a proposé un moyen pour déterminer l'image référence en se basant sur l'analyse de la moyenne du niveau de gris d'un pixel donné sur une séquence de plusieurs image le principe est : à chaque prise d'une nouvelle image de la séquence une nouvelle référence est construite. Pour avoir le niveau de gris de l'image référence il faut appliquer la relation suivante :

$$R(x, y, t) = \frac{1}{N} \sum_{i=t-N}^{t-1} I(x, y, t)$$

Avec :

- R : image référence présenté par (les coordonnées « x, y » et « t » la notion du temps).
- I : image actuel dans le temps.
- N : Nombre d'image dans la séquence.

L'approche présentée par Makarova besoin de sauvegarder N images successives dans la séquence ce qui est assez couteux. Elle présente aussi un autre problème lorsqu'il y a un changement de luminosité.

4. Le suivi d'objet en mouvement :

La détection du mouvement fournit des informations utiles pour le suivi d'objets. Il y a de nombreux efforts de recherche qui s'intéressent au problème de suivi. Les approches existantes peuvent être classées en deux catégories :

- Les approches qui se basent sur la frontière.

- Les approches qui se basent sur la région.

4.1. Les approches basent sur la frontière :

Ce type d'approches s'appuie sur les informations fournies par les frontières de l'objet. Il a été largement adopté dans le suivi d'objet car les caractéristiques basées sur la frontière (bords) fournissent des informations fiables qui ne dépendent pas du type de mouvement ou la forme de l'objet. Les algorithmes de suivi qui se basent sur ces approches utilisent deux méthodes :

- La méthode serpents [16] (model des contours actifs).
- La méthode des géodésiques contours actifs.

Le principe de ces modèles c'est de démarrer d'un contour initial et aller vers le contour recherché de façon itérative. Ces modèles sont très efficaces pour extraire les contours de l'objet avec une grande précision et ils sont très performants dans le cas de mouvements lents mais produisent de mauvais résultats lorsqu'ils adaptés à une quantité de mouvements importante.

4.2. Les approches basées sur la région :

Contrairement aux approches basées sur la frontière, les approches basées sur la région s'appuient sur les informations fournies par la région telle que les propriétés basées sur le mouvement en utilisant une technique d'estimation ou de segmentation de mouvement et de texture. Dans ce cas, l'estimation de la vitesse de la cible est basée sur la correspondance entre les régions de la cible associée à des instants de temps différents. L'utilisation de ces modèles présente la difficulté de suivre des limites dans le cas de mouvements/objets non rigides, mais augmente la robustesse dû au fait que les informations fournies par l'ensemble de la région soient exploitées.

5. La détection et le suivi en 3D :

La détection et le suivi de l'être humain est devenue l'un des sujets d'actualité pour de nombreuses applications telles que la vidéo-surveillance, les jeux vidéo ... etc.

Connaitre la position de l'être humain est une tâche nécessaire pour un robot guide qui est dans notre cas le robot B21r afin de fournir des services et interagir avec l'être humain de manière efficace.

Pour cela un système de détection et de suivi doit être élaboré pour exécuter les tâches suivantes :

- Détecter les personnes dans la scène.
- Présenter leurs positions 3D et leurs orientations par rapport à la camera.
- Suivre les déplacements des personnes

A cet effet il nous faut un matériel qui est capable de réaliser les tâches précédentes. Dans la littérature, on distingue deux types de matériels différents avec des mécanismes tout aussi différents :

- Détection et suivi à l'aide des marqueurs
- Détection et suivi sans marqueurs

5.1. Détection et suivi à l'aide des marqueurs :

La capture de mouvement en utilisant les marqueurs ou « motion caption » est une vieille technique. Elle consiste à capter un mouvement en temps réel pour l'appliquer en suite au modèle virtuel, il a eu des utilisations dans une variété de domaines des du militaires à la médecine. Elle est plus notable cependant pour son utilisation dans l'industrie du divertissement en particulier dans les jeux vidéo et le cinéma.

L'utilisation de « motion caption » communément appelé « mocap », peut générer des charges de données relativement rapides et après une courte période de fonctionnement livrer des très bons résultats.

5.2. Techniques utilisées dans la « mocap » :

La capture de mouvement avec la mocap se fait principalement avec deux types de technologies Ces technologies sont considérées comme optique car les cibles sont suivies par les caméras.

5.2.1. Détection et suivi avec marqueurs passifs :

Dans cette méthode les caméras sont mise en place pour suivre les réflexions de lumière venant de surfaces spéciales sur un costume ou le corps de l'acteur que se fait généralement avec des petites boules hautement réfléchissant.

Le processus est simple, les cameras émettant des lumières infrarouges qui sont renvoyés par les marqueurs aux caméras, ensuite, attachée un model 3D pour construire un personnage ou « avatar ». (Figures 15,16)

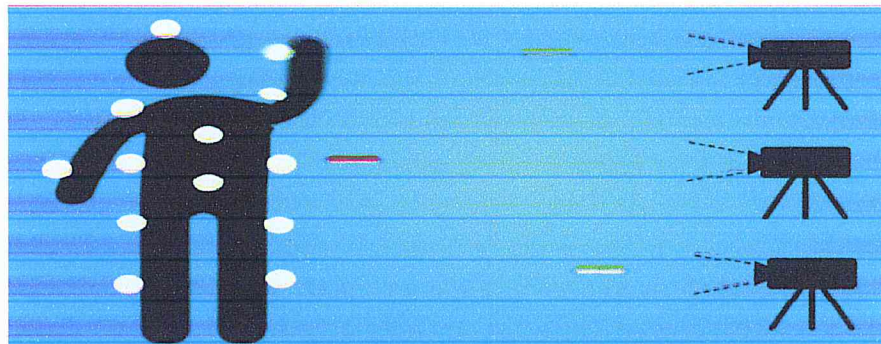


Figure 15 : Les marqueurs sous forme de boules et les cameras envoi

Le mouvement des avatars peut être enregistré dans un format particulier BHV [10] qui est un format d'animation de personnage et peut être traité dans un logiciel 3D comme Maya, 3dsMax, cinema4D Cette méthode est très sensible aux obstacles.



Figure 16 : Exemple réel de Andy-Serkis allias Gollum dans le Seigneur des Anneaux

5.2.2. Détection et suivi avec marqueurs actifs :

Cette méthode est fiable et elle est utilisée pour enlever le problème de la sensibilité aux obstacles. Contrairement à la détection et le suivi avec marqueurs passifs, l'acteur

dispose d'un costume avec des marqueurs LED (s'appellent aussi les capteurs gyroscopique) sur les surfaces de son corps a fin de crée le squelette virtuel.(Figure 17)

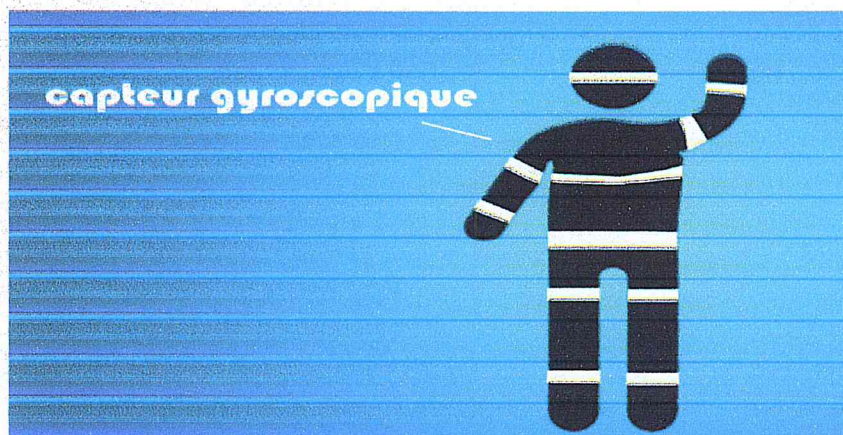


Figure 17 : Les capteurs led placé sus le corps de l'acteur

La figure ci-dessous montre le processus de création d'un avatar en quatre étapes :



Figure 18 : Exemple montre le processus pour crée un avatar en quatre étapes.

5.3. La Détection et le suivi sans marqueurs :

La détection et le suivi de l'être humain dans les images ou les vidéos est devenu un problème difficile à cause de la variation dans la pose, les conditions d'éclairage et la complexité des milieux, plusieurs méthodes ont été proposées.

D'après [17, 19,20] on voit que la recherche se base sur des images prises par des caméras simples ou les cameras RGB comme une caméra 'webcam', car elles capturent des images de façon naturelle comme ce que les yeux humains effectuent.

Certaines méthodes utilisent l'apprentissage statistique basé sur des caractéristiques locales par exemple : basé sur le gradient tel que HOG [17] et EOH [18] ... etc.

Bien que beaucoup de rapports ont montré que ces méthodes peuvent donner des résultats de détection humaine très précis, les méthodes basées sur les images RGB (ou RVB) rencontrent des difficultés à percevoir les formes des humains avec les différentes positions ou lorsque l'environnement est encombré. Ceci peut produire des augmentations des couts informatiques.

Pour cette raison, l'information de profondeur est importante pour donner à un robot la capacité de se déplacer et d'interagir avec son environnement.

Pour capter la profondeur, plusieurs techniques ont été proposé d'après [21] certaines utilisent des capteurs spéciaux pour créer une mesure physique de la profondeur, il Ya aussi des techniques qui calculent la profondeur en se basant sur des images couleurs par exemple : analyser les ombres ou le mouvement dans une scène, l'utilisation de deux caméras donne une vision stéréoscopique. La popularité de la vision stéréoscopique s'explique par analogie avec le système visuel humain et la disponibilité de caméras couleur à bas prix.

Cependant, les systèmes de vision stéréoscopiques ne sont capables que de calculer la profondeur de la scène pour un ensemble restreint de pixels correspondant à des zones de la scène ayant une forte structure locale, reconnaissable facilement d'une image à l'autre par exemple, il est impossible d'obtenir par stéréoscopie une information fiable sur la profondeur d'un objet de couleur unie, par exemple une feuille blanche qui occuperait le champ entier de la caméra.[22]

Des caméras 3D permettent de mieux capturer la profondeur d'une scène. Ces caméras 3D, dites actives, possèdent leur propre source de lumière pour calculer la profondeur. Par exemple, les caméras dites à temps de vol envoient une lumière visible ou

infrarouge et mesurent le temps entre l'émission d'une onde et la réception du signal réfléchi par la scène, à la manière d'un écho.

Dans notre travail on se base sur ce type de cameras, plus précisément la Kinect.

5.4. La Kinect :

Microsoft a lancé la Kinect qui est très facile à utiliser et simple à installer, le projet de la Kinect a commencé sous le nom « projet Natal » en 2009, elle a été destinée dans le premier temps aux joueurs sur xbox360, l'objectif c'est de remplacer la manette par exemple : si le joueur veut aller à droite dans le jeu il suffit juste de pencher son corps sur la droite ... etc.

La Kinect est dotée de deux caméras et de micros, comme la figure montre le premier capteur permet de capturer l'image en couleur tandis que la seconde permet de capturer la profondeur grâce à un capteur infrarouge. (Figure 19)

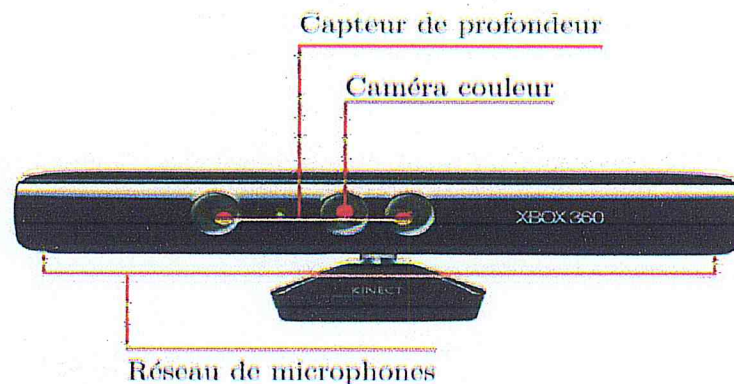


Figure 19 : image montre les principales parties de la Kinect (image face)

5.4.1. Les caractéristiques de la Kinect :

Le tableau suivant montre les caractéristiques de la Kinect.

Caractéristiques	propriétés
Le champ de vision	<ul style="list-style-type: none"> • Le champ de vision Horizontal : 57° (dégrée) • Le champ de vision vertical : 43° (dégrée). • La portée du capteur : 1.2m – 3.5m (pour avoir une bonne détection il faut être dans ce champ-là).
Le flux de données	<ul style="list-style-type: none"> • Capture de 30 images par seconde sur une résolution de 320X240 en 16bits de couleurs • Capture de 30 images par seconde sur une résolution de 640X480(VGA) en 32bits de couleurs • Flux audio sur une fréquence de 16kHz sur 16bits
Système de Reconnaissance Physique	<ul style="list-style-type: none"> • Jusqu'à 6 personnes et 2 joueurs actifs (4 joueurs actifs avec le SDK 1.0) • 20 articulations par squelette (varie par version du driver)

Tableau 1 : les caractéristiques de la Kinect

La figure ci-dessous montre le champ de vision de la Kinect.

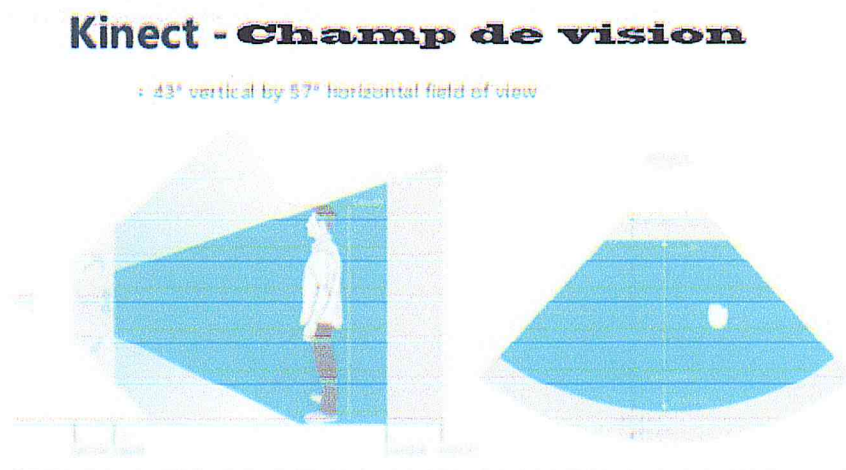


Figure 20 : champ de vision de la kinect

5.4.2. La capture de la profondeur :

Comme on a dit la Kinect possède un capteur infrarouge pour capter la profondeur, cette technologie a été introduit par la société primeSense le capteur se compose d'un émetteur de lumière infrarouge, et un récepteur qui permet d'obtenir une image qui représente les dégagements thermiques émis par l'objet observé, ce type d'images ne sont pas colorées, c'est les utilisateurs qui décident de rajouter les niveaux de couleurs en fonction de la température mesurée.

5.4.3. Utilisation de la Kinect :

L'Enterprise primeSense en 2010 a développé un driver sous le nom 'open Ni' pour utiliser le capteur de la profondeur, un outil très utile pour développer de l'application basée sur l'interaction naturelle comme le mouvement du corps humain, la voix... etc. Ce driver qui est open Source possède aussi un middleware sous le nom Nite² offrant un hôte léger, minimal charge CPU, et le support multiplateforme.

Ce middleware fournit l'application avec une API de contrôle utilisateur claire, que ce soit le contrôle à base de main ou un contrôle complet du corps. Les algorithmes utilisent la profondeur, la couleur, l'infrarouge et des informations audio reçues du dispositif matériel (Figure 21), qui leur permettent d'exécuter des fonctions telles que :

- la localisation de la main et le suivi.
- Un analyseur de scène (séparation des utilisateurs de fond)

Détection suivi des différents points dans le squelette humain

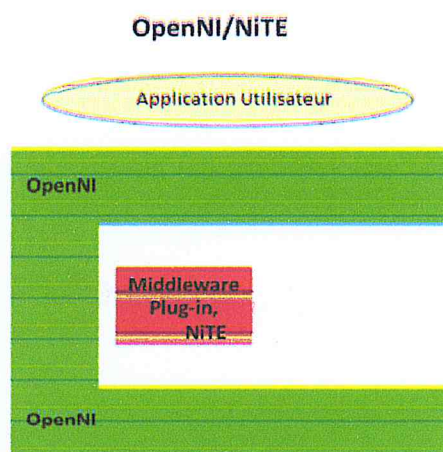


Figure 21 : Architecture du driver Open Ni

² <http://openni.ru/files/nite/>

En 2012 Microsoft a lancé la Kinect pour Windows avec un driver ‘SDK’ qui est un driver conçu pour le système d’exploitation Windows, le SDK de Windows offre des outils pour le développement en vision 3D et des fonctions de traitements meilleures que les autres concurrents mais ces fonctionnalités sont exploitées sous Windows seulement ce qui produit un problème lorsque on veut développer des applications pour d’autres plateformes notamment MacOS et linux. Le tableau suivant montre une simple comparaison entre OpenNI et le SDK Kinect :

openNi	SDK Windows
<ul style="list-style-type: none"> ➤ Introduit en 2010 par PrimeSense ➤ open Source ➤ multiplateformes ➤ utilisation de plusieurs langages programmation ➤ version instable. ➤ la position ‘PSI’. 	<ul style="list-style-type: none"> ➤ Introduit en 2012 par Microsoft ➤ close Source ➤ uni-plateforme ➤ utilisation de langages Microsoft ➤ version stable ➤ manque de la position ‘PSI’ détection directe

La figure ci-dessous montre la différence entre openni, SDK Windows et la technologie de MoCap :

Tableau 2 : montre la comparaison entre openni et SDK Windows

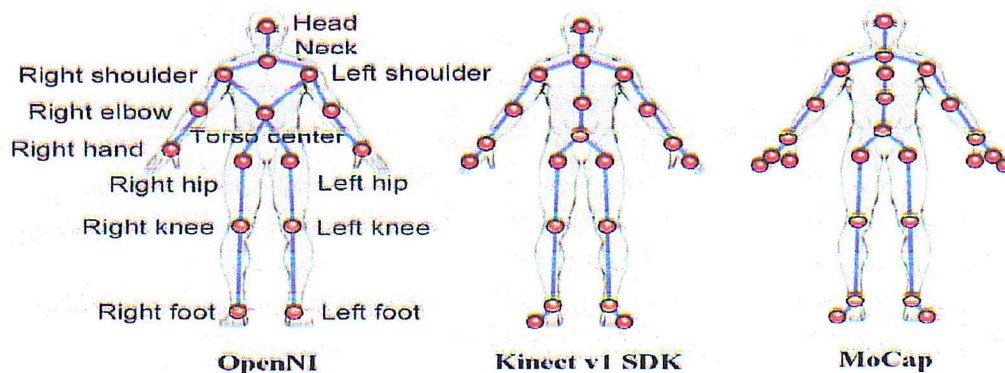


Figure 22 : la différence entre openni, sdk windows et MoCap

5.4.4. La détection et le suivi en utilisant la Kinect :

Dans l'article [21] un Framework a été proposé se base sur la notion du pipeline, pour augmenter la précision de la détection. Le principe est de fusionner efficacement les informations prévenantes des canaux images en niveau de gris et de profondeur, Dans le niveau de détection de chaque élément visuel des filtres basés sur la profondeur sont appliquées aux objets/homme détectés pour supprimer les fausses détections, Donc les informations de profondeur sont utilisées pour distinguer les différents types de scènes d'intérieurs.

Dans [22] les auteurs proposent une nouvelle méthode de détection humaine basée sur la carte de profondeur 3D, Un filtrage de pixels est utilisé pour réparer le plan de profondeur en raison de l'inexactitude des informations capturé par la Kinect, Ensuite un ensemble de données composé de cartes de profondeur avec diverses poses humaines est construits comme point de repère. A la fin ils ont utilisé l'algorithme SVM comme classifieur, L'approche proposée est comparé avec d'autres descripteurs existant qui utilisent le même classifieur, les résultats manifestent l'efficacité de l'approche proposée.

Dans [24], les auteurs ont présenté un système pour la perception 3D des humains en temps réel pour un robot automoteur. Leur approche permet de réduire le temps de calcul pour obtenir des performances en temps réel grâce à une combinaison unique de nouvelles idées et techniques établies. Ils introduisent un nouveau concept qui se base sur l'utilisation d'une cascade de détecteurs qui permettent de distinguer les humains des objets. Cette approche a donné une méthode robuste par rapport aux difficultés de l'environnement réel, comme les occlusions, les déplacements du robot, la disparition et la réapparition de l'être humain.

- **Approches basées sur OpenNI :**

La recherche pour une meilleure solution pour détection et suivi de l'être humain nous a conduit à trouver une API(package) sous le nom 'Openni_tracker' qui utilise la bibliothèque de OpenNI³ qui nous permet de faire une détection et un suivi à l'aide des informations de profondeur. L'avantage de cette API est qu'elle donne le squelette de l'être humain et les points dans le corps. L'utilisation de cette API avec ses algorithmes de détection et de suivi est très importante pour notre recherche.

³ <http://structure.io/openni>

Méthodologie :

Plusieurs recherches ont été faites pour réaliser un algorithme qui est capable de détecter et suivre l'être humain et construire un squelette, dans [25] A.Kar&al ont proposé une solution pour résoudre le problème de suivi du squelette d'un corps humain, en se basant sur un modèle bâton de squelette qui s'intéresse sur la partie supérieure du corps humain. Ils ont utilisé une variété de techniques de vision par ordinateur. Leur algorithme prend en entrée un ensemble de données RGB et les données de profondeur. A la fin, l'algorithme produit la silhouette de l'être humain représentée par 8 points.

Dans ce qui suit, nous allons donner le schéma général de l'API et les algorithmes utilisés pour réaliser la détection et le suivi de l'être humain en représentant la notion du temps réel. (Figure 23)

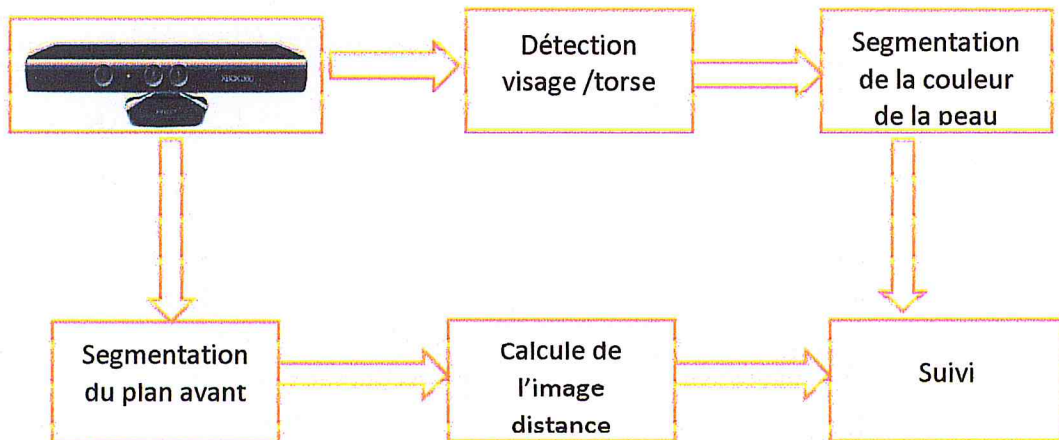


Figure 23 : l'architecture globale du suivi d'A.kar

❖ Image profondeur :

- Segmentation du plan avant :

Dans la segmentation du plan avant ils ont utilisé un seuillage sur l'image de profondeur pour extraire le premier plan de l'image, aussi ils ont éliminé le bruit cela est utile pour se focaliser uniquement sur le sujet dans l'image et de calculer la transforme de distance.

- Calcul de l'image de distance :

Le calcul de l'image de distance permet de créer une silhouette de l'être humain.

❖ Image basée sur RGB :

- Détection visage /torse :

Ils ont appliqué des classifieurs en cascade [26] pour détecter le torse et le visage de l'être humain .Chaque classifieur utilise une méthode pour classifier les régions de l'image.

- Segmentation de la couleur de la peau :

La peau est détectée par une projection dans l'espace de couleur HVS ce qui permet de récupérer les régions de la peau.

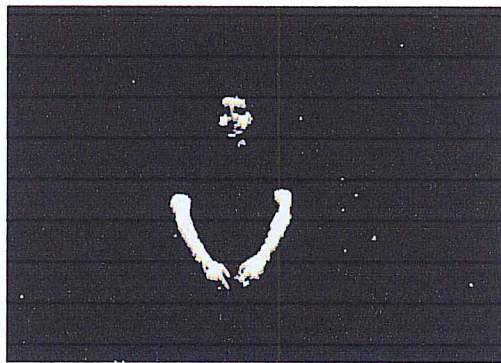


Figure 24 : montre la segmentation de la peau (la peau en couleur blanc)

❖ Le suivi :

Dans cette étape on trouve qu'ils ont présenté un modèle de squelette de l'être humain qui contient 8 points comme le montre la figure 25. Ils se sont basés sur les données trouvées dans (Nasa anthropométrie)⁴, un livre qui donne les mesures du corps humain comme le montre la figure suivante :

⁴ <http://msis.jsc.nasa.gov/sections/section03.htm>

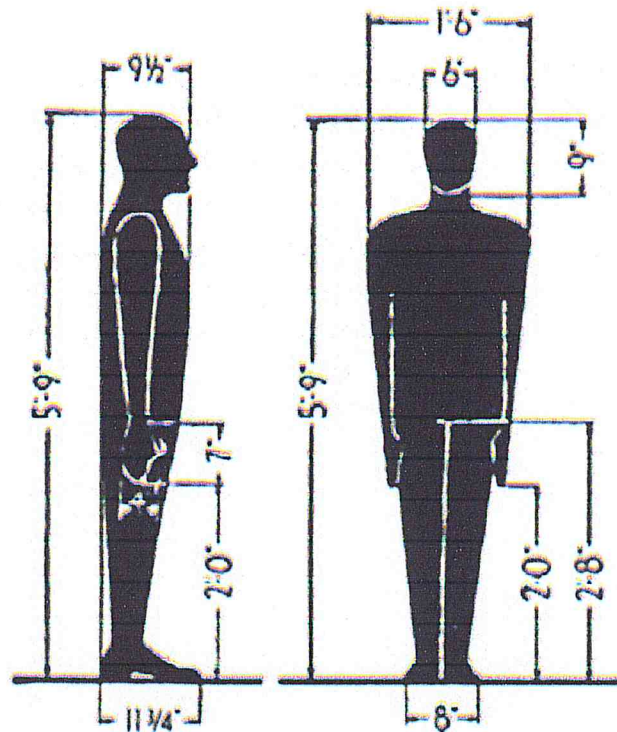


Figure 25 : les mesures du corps humain

Pour faire une mise en correspondance entre le squelette de l'être humain et l'image de distance, des suppositions sont faites sur les positions initiales de la tête et du cou (Figure 26), tel que la tête soit au centre du rectangle de détection du visage et le cou soit au milieu du côté bas du rectangle, de même pour les épaules, qui sont fixés au milieu des rectangles de détection du visage et du torse. L'estimation du coude se fait par la recherche dans un angle de pivot de l'épaule, de même pour l'estimation du poignet qui se fait dans un angle de pivot du coude. [27]

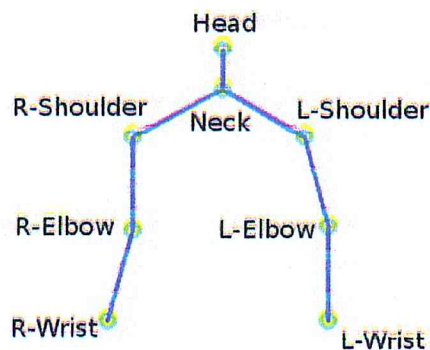


Figure 26 : les points du squelette de l'être humain

6. Conclusion :

A travers ce chapitre, nous avons donné une vision panoramique de la détection et du suivi de l'être humain.

Dans la première partie de ce chapitre, nous avons traité la détection et le suivi des objets mobiles et en mouvement sur différents points : changement d'image, approches basées sur la forme, approches basées sur la couleur, détection basée sur la correspondance de modèle, détection de Mouvement, détection de mouvement par segmentation, détection de mouvement par méthode de binarisation, détection par changement d'image ... etc.

Dans la deuxième partie, nous avons souligné la notion de profondeur en ce qui concerne la détection et le suivi en 3D en utilisant la Kinect comme support pour notre expérimentation.



Chapitre 3

Classification

1. Introduction :

Dans ce chapitre nous allons voir d'abord donner la définition de la classification et les différentes méthodes et algorithmes de classification en suite l'objectif de la classification, à la fin on optera pour un des algorithmes de classification pour l'appliquer dans notre solution.

2. Définition :

La classification c'est l'action de constituer ou de construire des classes, des groupes ou des catégories, les classes sont des ensembles d'individus ou d'objets qui possèdent des traits caractères communs (des traits caractères communs c'est-à-dire que ce sont des individus qui se ressemblent du point de vue de l'ensemble des caractères qui les Décrivent).

La classification automatique consiste à regrouper divers objets (les individus) en sous-ensembles d'objets (les classes). Elle peut être :

- Supervisée : les classes sont connues à priori, elles ont en général une sémantique associée.
- Non-supervisée (en anglais clustering) : les classes sont fondées sur la structure des objets, la sémantique associée aux classes est plus difficile à déterminer.

2.1. Classification Supervisée :

L'objectif de la classification supervisée est principalement de définir des règles permettant de classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets. Les méthodes s'étendent souvent à des variables Y quantitatives (régression).

On dispose au départ d'un échantillon dit d'apprentissage dont le classement est connu. Cet échantillon est utilisé pour l'apprentissage des règles de classement.

Il est nécessaire d'étudier la fiabilité de ces règles pour les comparer et les appliquer, évaluer les cas de sous apprentissage ou de sur apprentissage (complexité du modèle). On utilise souvent un deuxième échantillon indépendant, dit de validation ou de test.

Il existe plusieurs méthodes de classification supervisé, on cite :

- Arbre de décision
- Réseaux de neurone
- Support Vector Machine
- Régression logistique

comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts, l'objectif étant de permettre une extraction de connaissance organisée à partir de ces données.

Il existe plusieurs méthodes de classification non supervisée citons :

- **Des classifications appelées hiérarchiques** : pour le quelle on cherchera à construire un arbre hiérarchique pour voir comment s'organisé les objets ou les individus.
- **Des classifications de type méthode de partitionnement** : dont on va essayer uniquement de constituer des groupes d'individus qui se ressemblent et constituer une partition.

3. Indices de ressemblance :

La distance est indicée pour faire la ressemblance entre individus, il existe plusieurs types de distances citons :

- La distance euclidienne.
- La distance de Manhattan.
- La distance de Minkowski.
- La distance camberra.

3.1. La distance euclidienne :

C'est la distance entre deux points x et y.

La formule générale :

$$d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

Avec $m \in \mathbb{N}^*$, $x \in \mathbb{R}^m$ et $y \in \mathbb{R}^m$

3.2. La distance de Manhattan :

La formule générale :

$$d(x, y) = \sum_{i=1}^m |x_i - y_i|.$$

Avec $m \in \mathbb{N}^*$, $x \in \mathbb{R}^m$ et $y \in \mathbb{R}^m$

3.3. La distance de Minkowski :

La distance de Minkowski est une métrique dans un espace vectoriel norme qui être considéré comme une généralisation des deux distances

- La distance euclidienne.
- La distance Manhattan.

La formule générale :

$$d(x, y) = \left(\sum_{i=1}^m |x_i - y_i|^q \right)^{\frac{1}{q}}$$

Avec $m \in \mathbb{N}^*$, $x \in \mathbb{R}^m$ et $y \in \mathbb{R}^m$, $q \geq 1$.

3.4. La distance Canberra :

La distance Canberra est une mesure numérique de la distance entre les pairs de points dans un espace vectoriel.

La formule générale :

$$d(x, y) = \sum_{i=1}^m \frac{|x_i - y_i|}{|x_i + y_i|}$$

Avec $m \in \mathbb{N}^*$, $x \in \mathbb{R}^m$ et $y \in \mathbb{R}^m$

4. Algorithmes de classification :

Les algorithmes de classification sont des algorithmes qui permettent de classer des objets observés dans des classes appelée « clusters ».

Les objets d'une même classe doivent être similaires et les objets de deux classes différents doivent être distincts.

Nous citons l'algorithme de classification hiérarchique ascendante, et l'algorithme K-means qui est de la famille des algorithmes de partitionnements.

A. Classification hiérarchique ascendante :

La classification hiérarchique est une classification qui permet d'obtenir une hiérarchie c'est-à-dire une collection de groupes d'observations.

Elle s'intéresse à des tableaux de données individus en lignes et variables quantitatives en colonnes.

❖ **Objective de la classification :**

L'objective de la classification c'est de produire une arborescence qui met en évidence les liens hiérarchiques entre les individus ou entre des groupes d'individus.

La représentation sous forme d'arbre permet également de détecter un nombre de classes naturel dans une population.

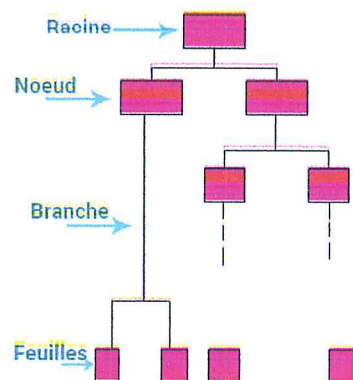


Figure 27 : Exemple d'une hiérarchie

❖ **Critères :**

Il faut pour réaliser une classification définir des mesures de ressemblance entre deux individus ces critères peuvent être :

- ❖ La Distance euclidienne.
- ❖ Indice de similarité

Il faut aussi définir une Mesure de ressemblance entre groupes d'individus

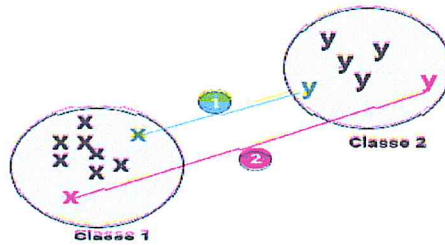


Figure 28 : Exemple de deux classes avec deux critères

Sur le petit schéma on voit deux groupes d'individus et la première mesure de ressemblance est le saut minimum.

On parle aussi de :

- Lien simple : c'est la distance minimum entre deux groupes qui égale à la plus petite distance entre un élément du premier groupe et un élément du second groupe (dans l'exemple le lien simple est représenté par '1')
- Lien complet : contrairement au lien simple, le lien complet est la plus grande distance entre un individu du premier groupe et un individu du second groupe (dans l'exemple le lien simple est représenté par '2')

Il existe plusieurs mesures de ressemblances entre individus, et plusieurs mesures de ressemblance entre groupes d'individus, le choix de la mesure modifie la classification que l'on obtient.

Donc selon les données, on utilisera certaines distances entre individus et certaines mesures de ressemblance entre groupes d'individus.

B. Algorithme de partitionnement k-means:

L'algorithme de partitionnement k-means est un algorithme de classification non supervisée, son objectif c'est de créer un partitionnement des objets dont les objets d'une même classe soient similaires et les objets de deux classes différents soient dissimilaires.

Le principe de cet algorithme est de définir 'Q' centroïdes pour chaque classe 'C', le choix des centres initiaux est réalisé sur une base d'un tirage aléatoire sans remise de 'k' individus à partir de la population à classer, la partition des classes est modifiée à chaque affectation d'un individu i de I . [28]

La formule de calcul de distance de l'algorithme se définit par :

$$T = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N d(X_i, X_j)$$

Dans :

D : distance

X : un ensemble de points.

La fonction donne :

$$T = \frac{1}{2} \sum_{k=1}^k \sum_{c(i)=k} (\sum_{c(j)=k} d(X_i, X_j) + \sum_{c(j) \neq k} d(X_i, X_j))$$

En note :

$W(C) = \sum_{c(j)=k} d(X_i, X_j)$: la distance dans la classe

$W(B) = \sum_{c(j) \neq k} d(X_i, X_j)$: la distance entre les classes

La formule devient :

$$T = \frac{1}{2} \sum_{k=1}^k \sum_{c(i)=k} (W(C) + W(B))$$

❖ Étapes de l'algorithme :

L'algorithme de k-means se déroule en quatre étapes [28] :

- Choisir k objets pour former k clusters.
- Réaffecter chaque objet O au cluster c_i de centre M_i talque distance entre O et M_i est minimal
- Recalculer M_i de chaque cluster
- Aller à 1 » étape 2 si on vient de faire une affectation

❖ L'arrêt de l'algorithme [28] :

L'algorithme s'arrête lorsque :

❖ **L'arrêt de l'algorithme [28] :**

L'algorithme s'arrête lorsque :

- Deux itérations successives conduisent à une même partition.
- Lorsqu'on fixe un critère d'arrêt tel que le nombre maximal d'itération.

Les inconvénients de cette méthode de classification sont que la classification finale dépend du choix de la partition initiale, et aussi il faut connaître le nombre de classes a priori.

5. Conclusion :

La classification hiérarchique et la méthode des "k-means" sont deux méthodes itératives et ont des principes différents. Pour cette dernière raison qu'il est difficile de comparer entre l'output des deux méthodes appliquées à un même input.

Ces deux méthodes se caractérisent par le fait que l'une complète l'autre. En effet, pour mener une classification en utilisant la méthode des "k-means", il serait préférable de commencer par lancer une classification sur un échantillon plus réduit que l'échantillon initial. Ensuite, on dégage le nombre de classes qui seraient utilisées pour l'affectation des individus aux classes les plus proches.

C'est l'une des propriétés de ces méthodes qui leur permet d'être les méthodes de classifications les plus utilisées pour la segmentation de la population.



Chapitre 4

Conception

1. Introduction

La Détection et le suivi à l'aide d'une caméra Kinect nous a permis de lire plusieurs paramètres notamment la position de l'être humain et aussi le suivi de ses mouvements.

Ces données peuvent être utilisées pour construire des classifications de personnes pour les utiliser sur le robot B21r.

La détection de personnes est un sujet particulièrement difficile, en raison notamment de la grande variabilité d'apparences et de situations possibles comme :

- La grande variabilité de l'apparence des êtres humains, ainsi que de leurs vêtements.
- L'articulation du corps humain (bras, jambes, torse).
- Les occultations par des objets (meublier urbain par exemple).
- Les occultations par d'autres personnes et phénomènes de foule.

Généralement, la détection doit s'effectuer dans des conditions difficiles et en environnement non contraint, en utilisant du matériel de prise de vue fournissant des images de faible qualité : caméras de vidéosurveillance de faible résolution, caméras embarquées dans un véhicule, dans un robot, etc.

Pour une détection plus avancée (intelligente) des personnes, (ex : détection de groupe de gens), il faut avoir recours à une classification des personnes détectées.

Détection actuel :

À partir des années 2000, le domaine bénéficie des avancées effectuées en détection de visage et notamment de la méthode de Viola et Jones[26], qui est étendue en 2005 à la détection de personnes en utilisant le mouvement

La méthode permet une détection plus générique, ne nécessitant pas d'information a priori sur la structure de la scène, et donne de bons résultats avec un temps d'exécution proche du temps-réel. Poussée notamment par les campagnes d'évaluation internationales (ex. PASCAL VOC), la recherche se concentre sur ce type de méthode de détection générique. En 2005, des chercheurs de l'INRIA proposent les histogrammes de gradient orienté (HOG), ont d'excellents résultats en font rapidement une méthode standard. En 2008, des chercheurs de l'Université Rutgers utilisent un descripteur construit comme une matrice de covariance, qui permet d'obtenir des performances encore meilleures

Également en 2008, des chercheurs de l'université de Pékin utilisent avec succès les motifs binaires locaux (LBP), un type de caractéristiques qui s'était révélé efficace en détection de visage.

Pour un robot guide muni d'une caméra RGB-D (Kinect), la perception de l'environnement doit prendre en compte les facteurs liés aux comportements sociaux habituels, par exemple ne pas franchir un couple ou un groupe de personnes, un suivi d'un groupe de gens, un comportement proche de l'être humain, la détection de personnes ne suffit pas.

Alors comment peut-on améliorer cette détection des personnes ? Quel sont les caractéristiques qu'il faut prendre ?

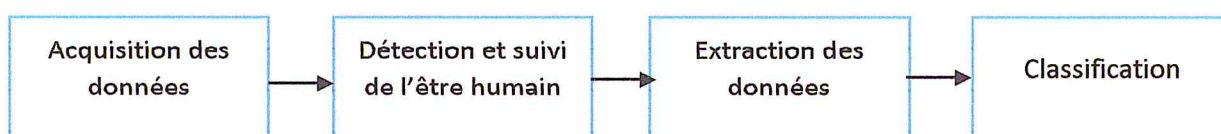
La réponse à cette question pourrait permettre de fournir une intelligence à cette détection, en conséquence une intelligence au robot lui-même.

La présente étude a donc pour but d'offrir une classification des personnes détectées par rapport à un ou plusieurs attributs notamment le nombre, la direction, la vitesse... etc.

2. Conception :

Dans cette partie on va traiter la conception de l'application à réaliser pour ce faire, on a divisé le travail en quatre parties.

Le schéma suivant montre les quatre étapes :



Dans ce qui suite on va détailler chaque étape.

2.1. Acquisition des données :

Dans cette première partie on va aborder la notion d'acquisition des données mais avant on doit définir quel le type de données capturé par la Kinect.

On distingue deux types de données :

A. Image couleur classique :

- La reconnaissance d'objets et de personne passe par l'analyse des pixels...

- La valeur des pixels dépend des conditions de prise de vue : éclairage, ouverture, modification des couleurs par l'appareil, etc...
- Impossibilité de connaître facilement le contenu d'une image, de savoir où commence et où finit un objet, etc...

B. Image en profondeur :

- L'intensité des pixels est proportionnelle à la distance de l'objet considéré avec la caméra.
- L'intensité des pixels est indépendante de l'éclairage
- Possibilité de localiser avec précision les objets dans la scène, où ils commencent / finissent, etc...
- Reconstruction 3D de la scène.
- Image en profondeur permet d'appliquer des algorithmes de suivi plus efficace que sur une image classique (Librairies performantes):
- Suivi des personnes
- Suivi des articulations du corps

A cet effet on trouve que L'image en profondeur est plus facile à analyser pour la machine qu'une image en couleur classique.

La figure ci-dessous montre la différence entre l'image couleur et l'image en profondeur .

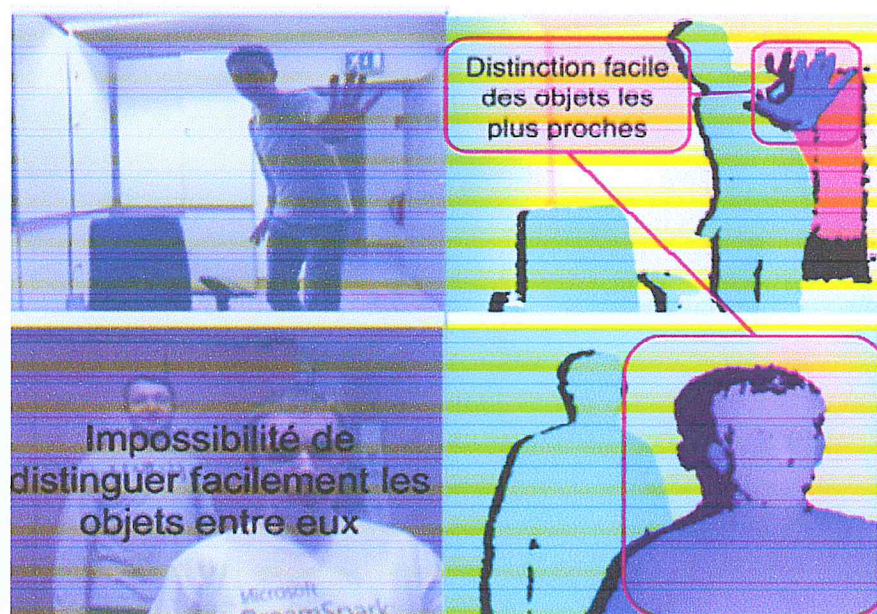


Figure 29 : différence entre image simple et image profondeur

Outils utilisables :

Nous avons le choix entre Trois technologies :

- OpenNI (OpenNI + NiTE + SensorKinect)
- Open Kinect (libfreenect)
- SDK Windows

Nous avons choisi OpenNI car cet outil était plus documenté. Il contient des applications natives qui nous intéressaient, notamment le suivi du squelette du joueur (ou Skelton tracking), non existant sur OpenKinect.

TRAVAIL SUR L'IMAGE EN PROFONDEUR :

Comme on a dit précédemment on travaille sur OpenNI, en utilisant l'une de ses bibliothèques **SimpleOpenNI** on peut extraire les informations de la Kinect en utilisant les fonctions de cette dernière.

Dans ce qui suite on présente les différentes étapes pour formaliser les données :

A. DepthImage (DI) :

La première étape est d'extraire l'image de profondeur de la scène, la fonction **depthMap ()** Renvoie un tableau-1D (liste) d'entier contenant les différentes valeurs des profondeurs mesurées selon la grille IR.

La figure ci-dessous montre un exemple d'une DepthMap

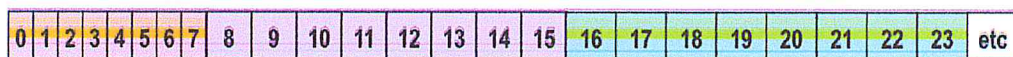


Figure 30 : exemple d'une DepthMap

Pour avoir la depthImage il faut tout simplement organiser la depthMap dans une matrice selon la résolution de l'image, par default c'est 640*480

La figure ci-dessous montre un exemple d'une DepthMap organisé

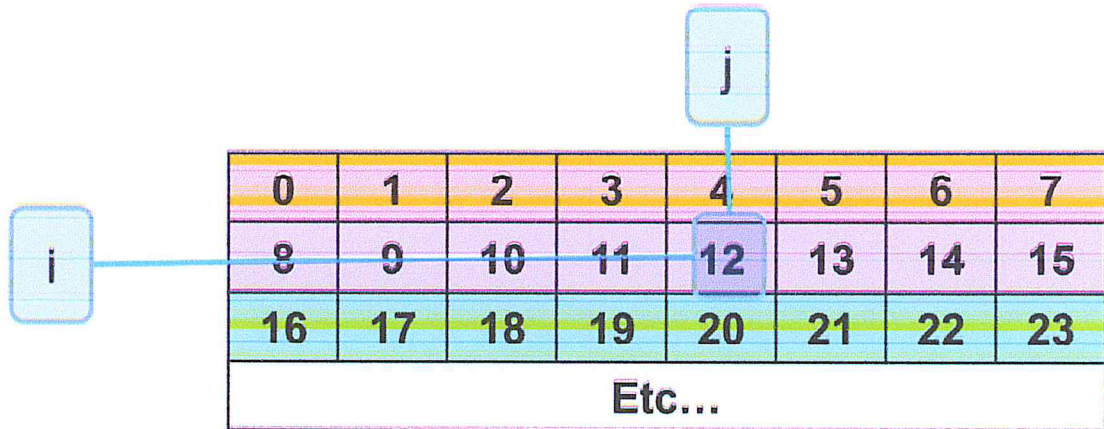


Figure 31 : exemple d'une DepthMap organisé

B. Coordonnes cartésiens :

Après avoir réalisé l'acquisition des images, on doit réaliser un traitement dessus.

Pour appliquer le traitement sur une depthImage, nous allons simplement calculer la distance moyenne Entre la Kinect et chacun des pixels de l'image (en ignorant les pixels pour lesquels nous n'avons pas de mesure de profondeur).

Le calcul de la distance moyenne est réalisé en deux étapes :

- En premier lieu, les coordonnées « projectives » (colonne, rangée, profondeur) de chaque pixel sont converties en coordonnées cartésiennes. Le système cartésien a son origine au niveau de la Kinect avec l'axe X pointant vers sa droite, l'axe Y vers le haut et l'axe Z vers l'avant OpenNI réalise cette conversion en allant récupérer les paramètres de calibration enregistrés en usine dans la Kinect. Tout est alors prêt pour calculer la distance entre chaque pixel et l'origine du système cartésien dans le but d'obtenir la moyenne. La figure ci-dessous montre les coordonnées cartésiennes x,y,z de la kinect.

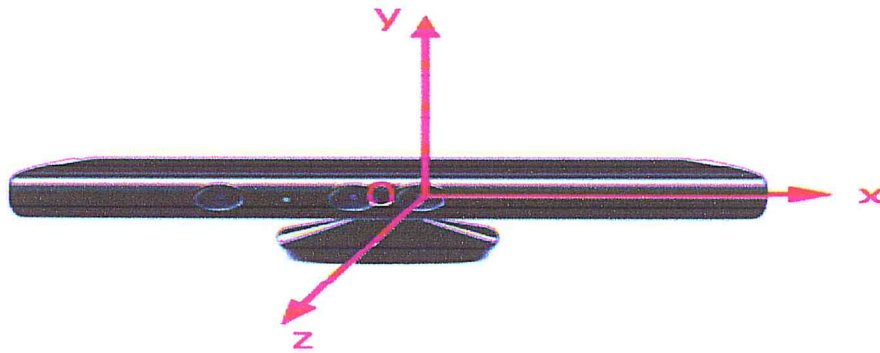


Figure 32 : les coordonnées x,y,z

- Après la conversion des pixels en coordonnées cartésiennes, le programme calcule la distance euclidienne de chaque pixel à l'origine (la Kinect elle-même)

2.2. Détection & suivi de l'être humain :

L'interface d'OpenNI qui permet de détecter les personnes situées devant une caméra s'appelle un générateur d'utilisateur (UserGenerator). De base, ce générateur est capable de détecter les différents utilisateurs situés devant la caméra et de fournir une image où chaque pixel est annoté avec l'identifiant de l'utilisateur dont il fait partie. OpenNI propose plusieurs extensions au générateur d'utilisateur pour interpréter les mouvements humains, dont notamment :

- L'extension « squelette » qui permet d'obtenir la position de différentes articulations du corps humain.
- L'extension de détection de pose qui permet de détecter quand un utilisateur se place selon une pose précise (par exemple, les bras en l'air).

La seule implémentation disponible du générateur d'utilisateur est celle développée par la société PrimeSense : le middleware propriétaire NITE.

Les données générées étant dépendantes du contenu de la scène, OpenNI fournit un mécanisme de fonction de rappel, appelée aussi fonction de callback, pour signaler au programme différents événements comme l'apparition d'un nouvel utilisateur.

La détection de pose est une extension du générateur d'utilisateur, La liste des poses détectables peut être obtenue avec la fonction `xnGetAllAvailablePoses`.

Les poses sont représentées par une chaîne de caractères.

La figure ci-dessous Montre la pose « Psi » reconnaissable par le middleware NITE :

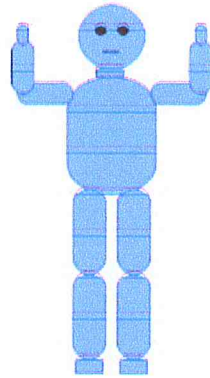


Figure 33 : la position 'PSI'

La capacité « squelette » permet de récupérer la position de différentes articulations du corps humain, à savoir :

- La tête
- Le cou
- Le torse
- Les épaules
- Les coudes
- Les mains
- Les hanches
- Les genoux
- Les pieds
- Les clavicules
- Les poignets
- La taille
- Les chevilles
- Le bout des doigts

Le middleware NITE ne fournit l'information de position que pour les articulations mentionnées, On peut également spécifier un sous-ensemble d'articulations à suivre, qu'il s'agisse de toutes les articulations, uniquement des membres supérieurs, uniquement des

membres inférieurs ou uniquement de la tête et des mains. Dans notre cas, nous voulons seulement la partie significative de l'humain qui est le torse.

Avant d'obtenir ces informations, il faut que la librairie se calibre pour l'utilisateur. La calibration peut être automatique (comme depuis la version 1.5 de NITE) ou dépendante d'une pose particulière (précédemment, NITE demandait la détection de la pose « Psi » pour calibrer le squelette).

La fonction `xnGetSkeletonJointPosition` permet de récupérer la position d'une articulation sous la forme d'une structure `XnSkeletonJointPosition`.

Cette structure contient le champ `position` de type `XnPoint3D` qui donne les coordonnées cartésiennes de l'articulation.

2.3. Extraction des données :

Dans cette partie on va parler du cheminement de données, le type et le contenu, pour cela les données doivent passer par trois phases.

Publication :

La première et Seconde étape de la conception sont entièrement assurées par un package open source de type ROS sous le nom de « OPENNI_TRACKER » écrit en c++, dans le quel nous avons ajouté des instructions au code qui nous permette de publier un message ROS dont les propriétés sont :

- Type : ros message
- Nom : pos
- Taille : 1000 octet
- Fréquence : 30 Hz

Le message contient toutes les informations concernant les personnes détectées :

- Identificateur
- Position par rapport aux axes «X, Y, Z »

La figure suivante montre les différents composants pour réaliser la publication.

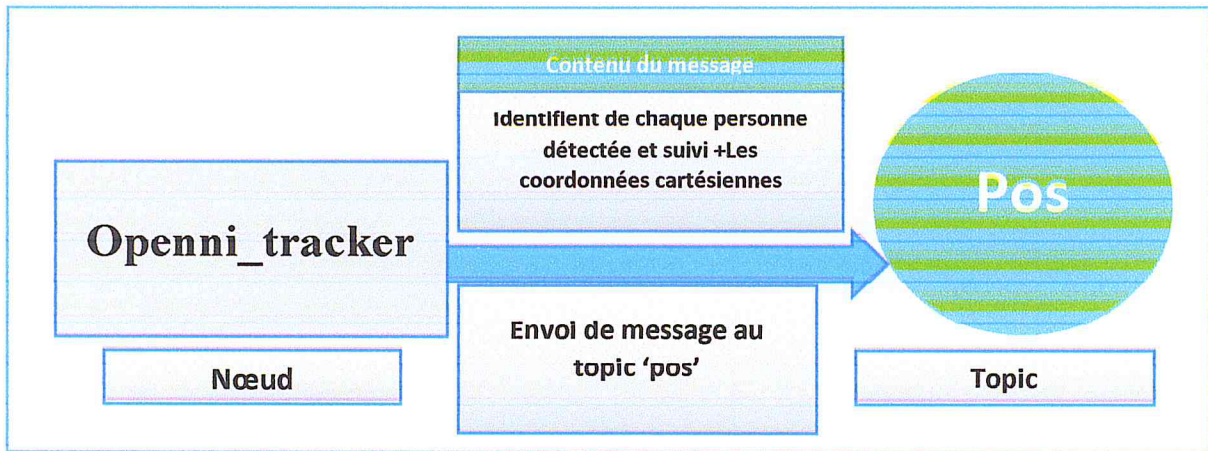


Figure 34 : les différents composants pour publier un message

Les messages sont publiés à une fréquence de 30 Hz soit l'équivalent de 30 messages par seconde.

Réception :

Pour la partie réception des données nous avons ajouté un nouveau nœud ROS qui a pour nom « classification » écrit en python, ce dernier nous permet de recevoir les messages envoyés par le package précédent « OPENNI_TRACKER », à la même fréquence soit 30 Hz

De manière générale, l'extraction des données se fait en premier ordre par une publication des messages via le package « OPENNI_TRACKER », en second ordre elle se fait par une réception via le nœud « classification ».

La figure suivante montre les différents composants pour réaliser la réception des données.

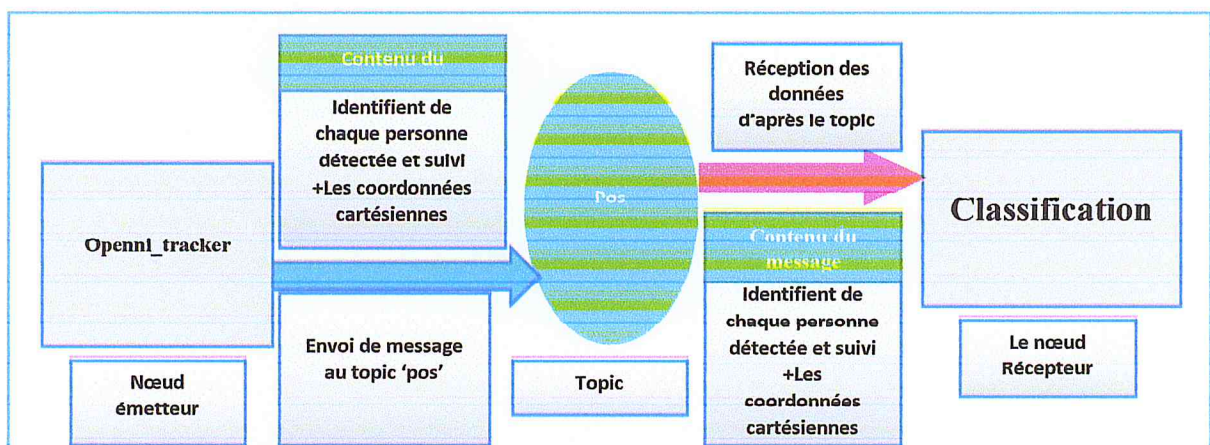


Figure 35 : les différents composants pour la réception d'un message

2.4. La classification :

Dans cette phase on va entamer la partie concernant la classification des informations reçues du package <Openni_tracker>, pour cela il existe une multitude de méthode de classification, on a sélectionné deux types :

- La classification par méthode hiérarchique ascendante
- K-means

Après étude de ces deux algorithmes de classification, on a adapté la classification par méthode hiérarchique ascendante avec un seuil de similarité qui est dans notre cas la distance euclidienne.

aussi on a évité de prendre la méthode k-means suite à la difficulté de fixer le nombre de classes a priori, également la méthode probabiliste a besoin d'une base d'apprentissage ce qui est loin de notre cas.

On a modifié l'algorithme de la méthode hiérarchique ascendante en l'ajustant sur notre approche, comme le montre l'organigramme suivant.

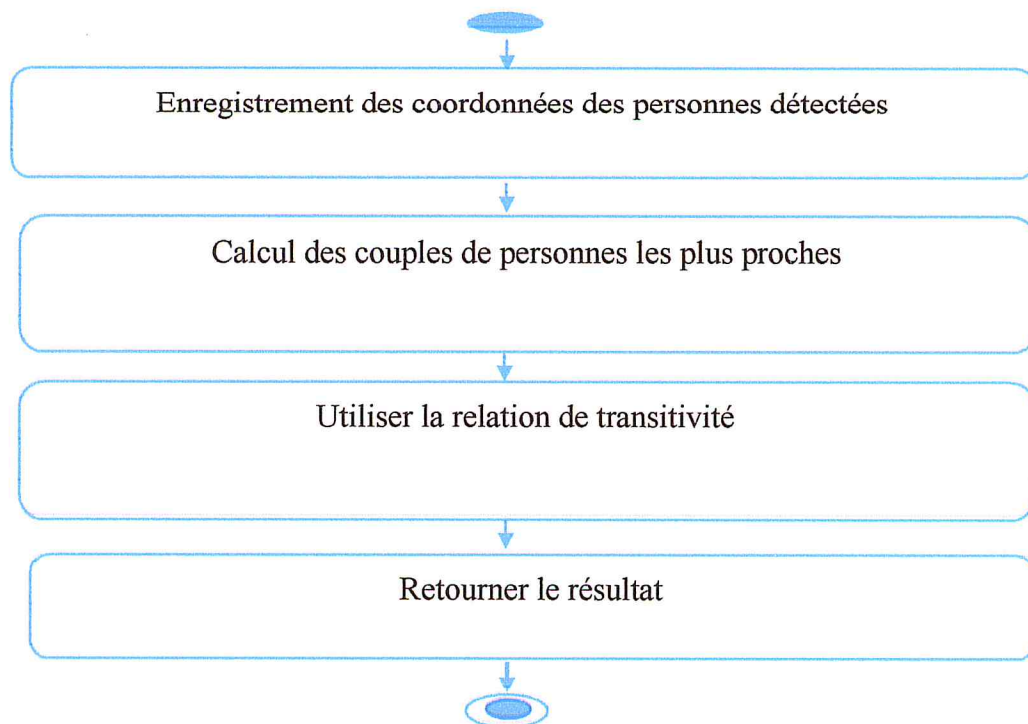


Figure 36 : Fonctionnement de l'approche

La figure suivante présente le pseudo algorithme de la fonction principale :

Algorithm 1 Classification

Requires: *id user X position Y position Z position*

Ensure:

- Update()* \\ *mise a jour des coordonnees de personnes dans le champ de vision.*
- generate-couple()* \\ *calcul des couples de personnes les plus proches.*
- relation-transitivit()* \\ *utilisation de relation de transitivité.*
- return-liste()* \\ *retourner le resultat finale.*

Figure 37 : pseudo algorithme de la fonction principale

Nous allons détailler chaque étape de la fonction dans ce qui suit

Enregistrement des coordonnées des personnes détectées :

La première étape consiste à créer une fonction qui nous permet d'enregistrer les coordonnées reçues qui sont de type cartésienne plus l'identifiant qui est attribué à chaque personne.

La figure suivante montre les étapes pour réaliser le premier traitement.

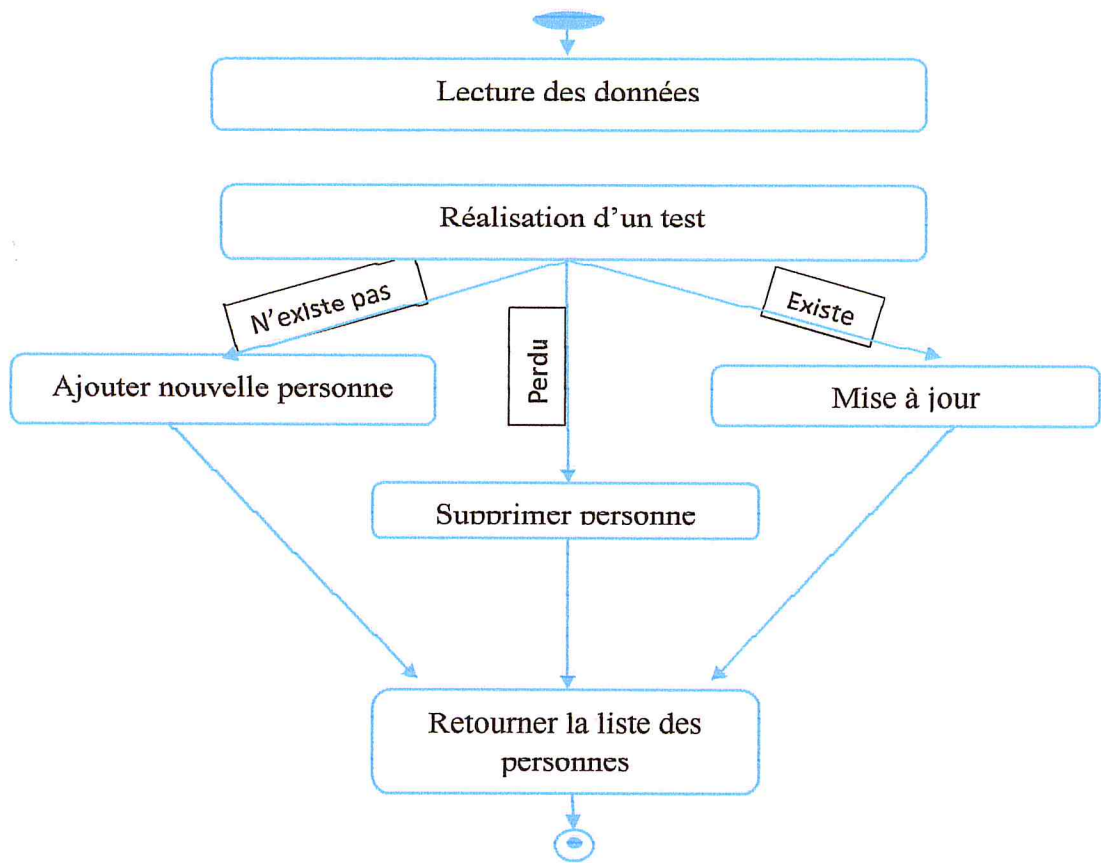


Figure 38 : Enregistrement des coordonnées

Après la lecture des données le programme effectue trois tests :

- Si la personne n'existe pas on ajout la personne.
- Si la personne existe on met à jour les coordonnées
- Si la personne sort du champ de vision on la supprime

La figure suivante montre le processus et le résultat de la 1^{ère} étape

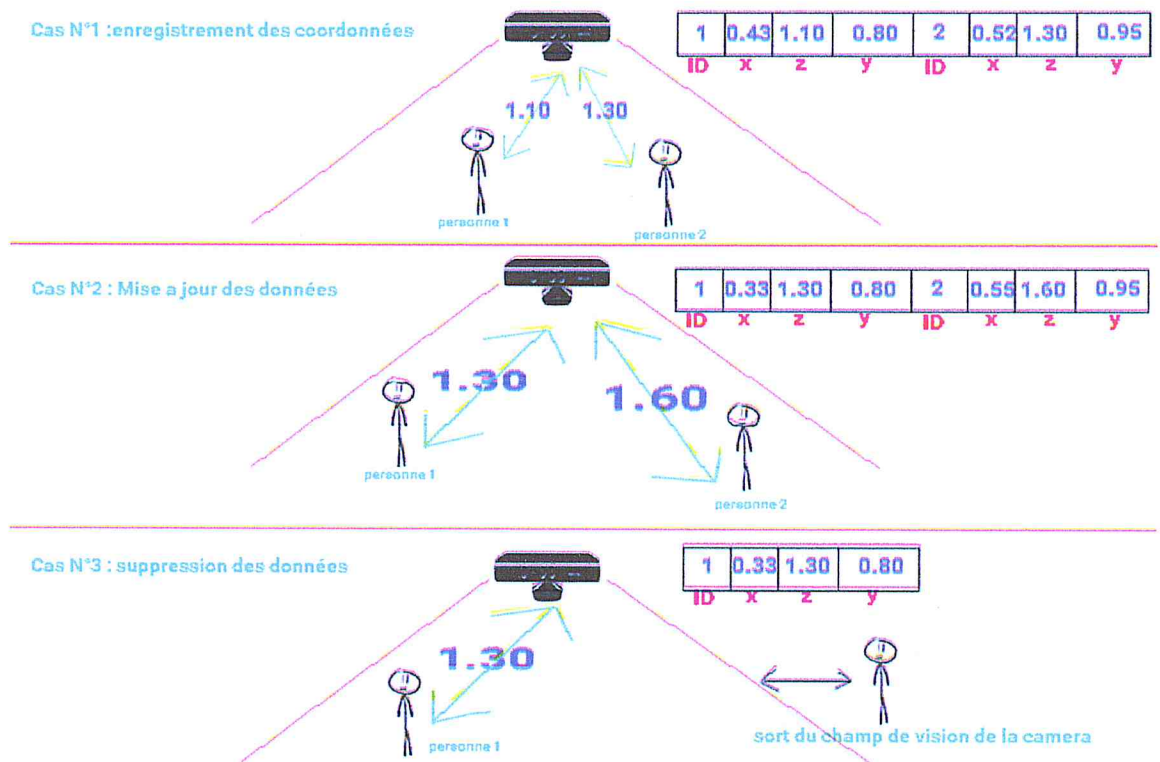


Figure 39 : le processus de la 1^{ere} étape

Calcul des couples de personnes les plus proches :

La deuxième étape consiste à comparer les distances (euclidienne) entre deux personnes en parcourant toute la liste, on choisit les couples qui possède une distance inférieure par rapport à un seuil définit.

La figure suivante montre les étapes pour réaliser le deuxième traitement.

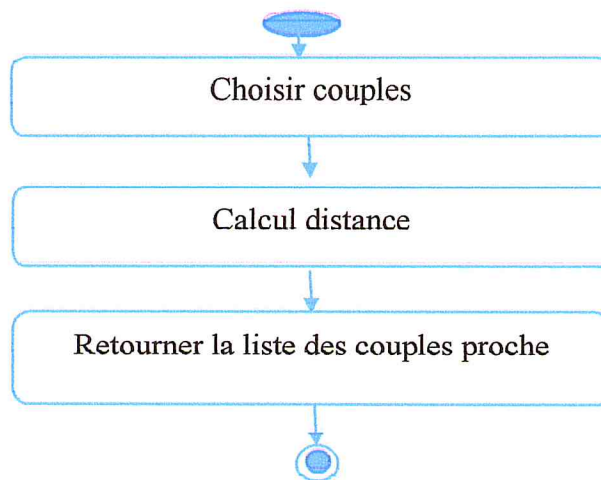


Figure 40 : Calcul des couples de personnes les plus proches

La figure suivante montre le processus et le résultat de la 1^{ère} étape

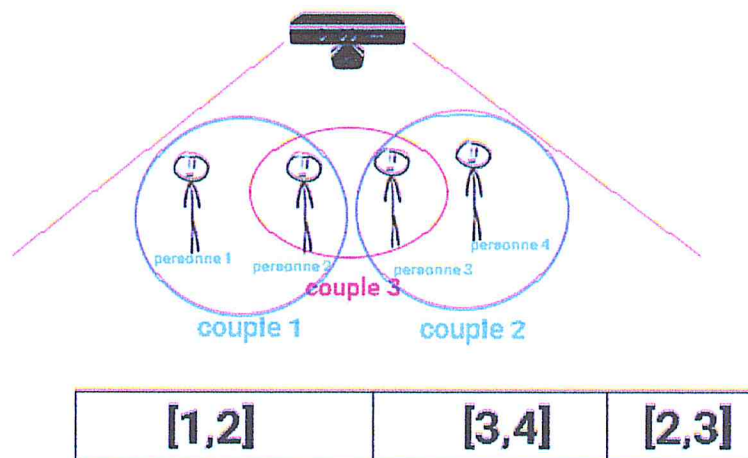


Figure 41 : le processus de la 2eme étape

Utiliser la relation de transitivité :

À ce niveau on a divisé le processus en quatre parties :

- La première partie recherche les personnes qui n'appartiennent à aucun couple,
- En second on applique la relation de transitivité entre les couples pour déterminer les groupes,
- En troisième lieu, on recherche les couples qui n'ont aucune relation de transitivité, enfin on retourne la liste des groupes.

La figure suivante montre les étapes pour réaliser le troisième traitement.

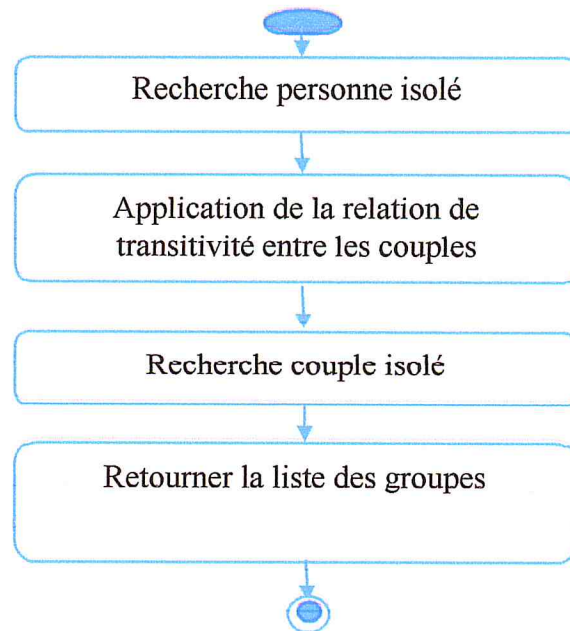


Figure 42 : Utilisation de la relation de transitivité

Pour mieux comprendre la définition de chaque classe est nécessaire :

- ✓ La personne isolée : est une personne qui n'a aucune personne proche d'elle (proche c.-à-d. elle ne dépasse pas le seuil définit)
- ✓ Le couple isolé : est un couple qui n'a aucune personne proche de ce dernier.

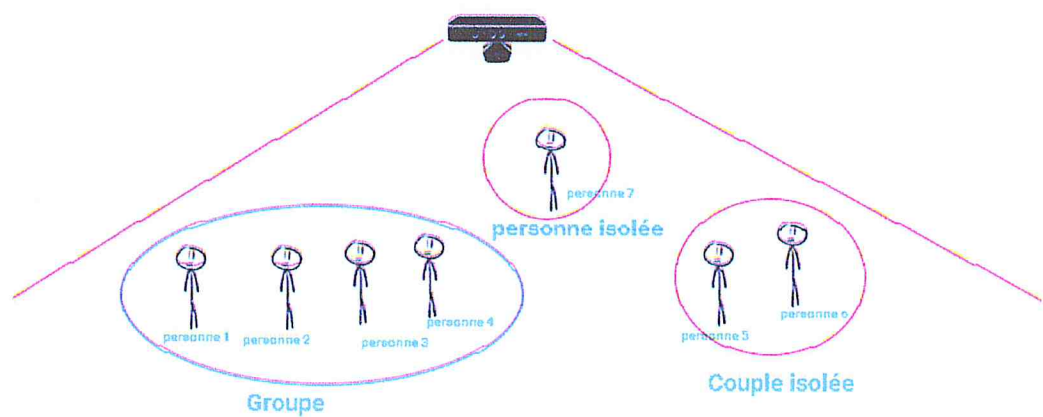


Figure 43 : le processus de la 3eme étape

Retourner le résultat :

Dans cette dernière étape le travail consiste à regrouper les personnes solitaires, les couples solitaires, et les groupes issus de la relation de transitivité dans une même et unique liste.

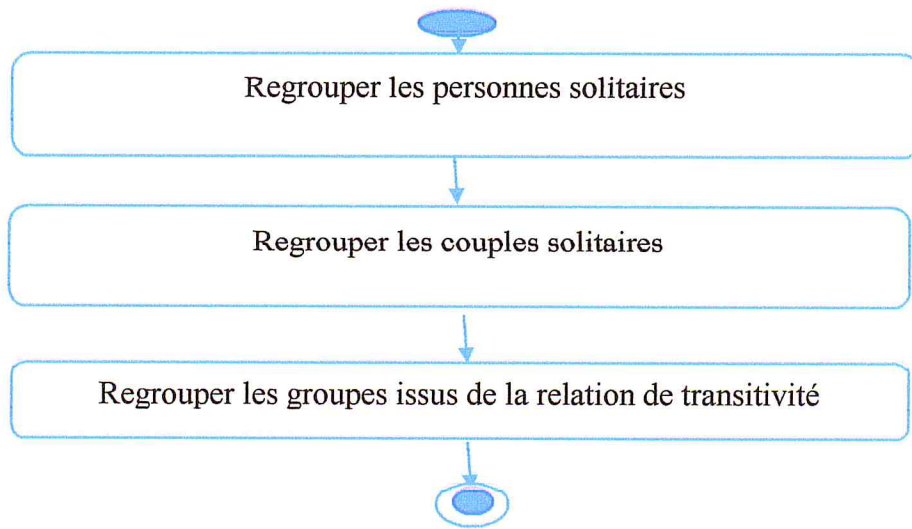


Figure 44 : Retour des résultats

La figure suivante montre le processus et le résultat de la 4^{ème} étape :

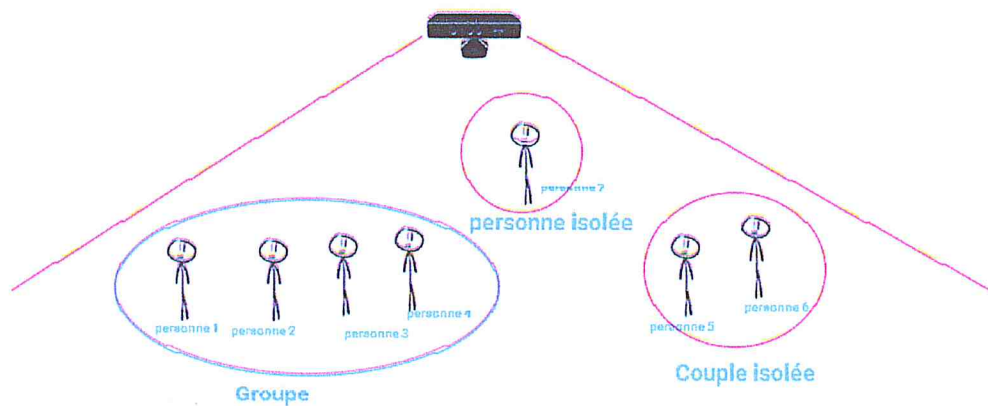


Figure 45 : le processus de la 4^{ème} étape

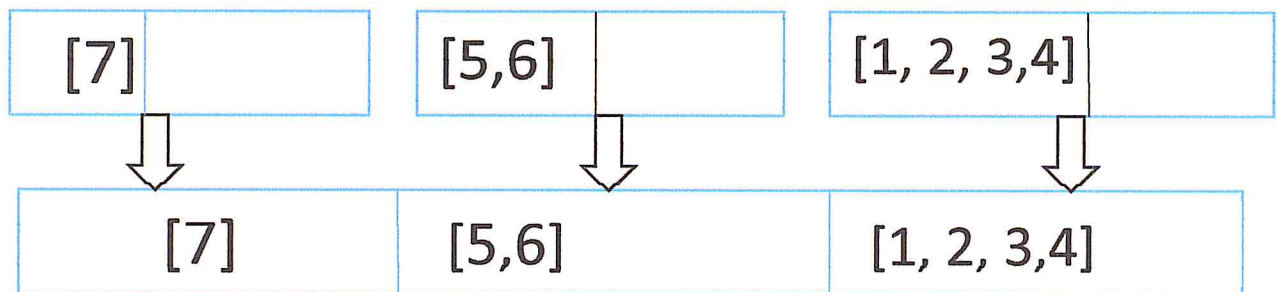


Figure 46 : le résultat final

3. Conclusion :

Dans ce chapitre on a présenté le schéma général du travail demandé.

Par la suite décrit notre solution via un schéma général, puis en expliquant les détails de chaque phase, En fin, nous avons terminé par présenter la solution et adapter cette méthode sur le robot B21r.

Dans le prochain chapitre, nous présentons les résultats et l'ensemble des tests effectués sur le robot .Ensuite, nous allons utiliser des benchmarks pour tester les performances et la fiabilité de notre approche.



Chapitre 5

Réalisation et tests

1. Introduction :

Dans ce dernier chapitre, nous montrons l'architecture de notre système robotique expérimental. Ensuite, nous allons montrer les différents tests que nous avons choisis pour valider la solution proposée.

Les tests seront composés de deux parties :

- Tests effectués en simulation pour éviter tout dégât matériel.
- Tests effectués sur la plate-forme robotique expérimentale réelle.

Enfin nous présentons les résultats des tests en matière performance, temps de réponse, et de fiabilité de notre approche.

2. Architecture globale :

L'architecture globale, est divisée en trois composants :

- L'ordinateur placé sur le robot.
- La Camera Kinect RGB-D.
- Le robot B21R

La camera Kinect sert d'entrée d'information sur l'ordinateur.

Nous avons intégré notre programme de détection, suivi, et classification l'être humaine, dans l'ordinateur.

Le Robot B21R représente une sortie interprété par exemple de mouvement ou de suivi

Cela dit, nous allons présenter dans ce qui suit en bref l'architecture globale du système Robotique, avant de présenter les résultats obtenus.

La figure ci-dessous montre l'architecture globale.

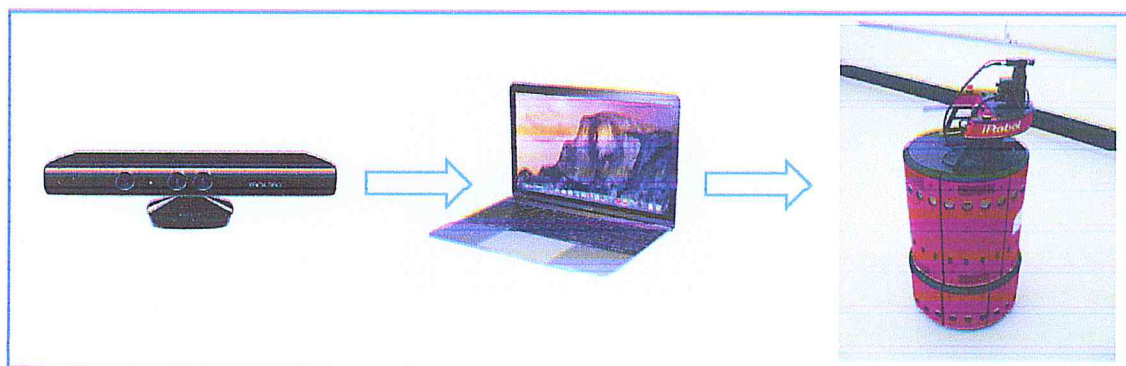
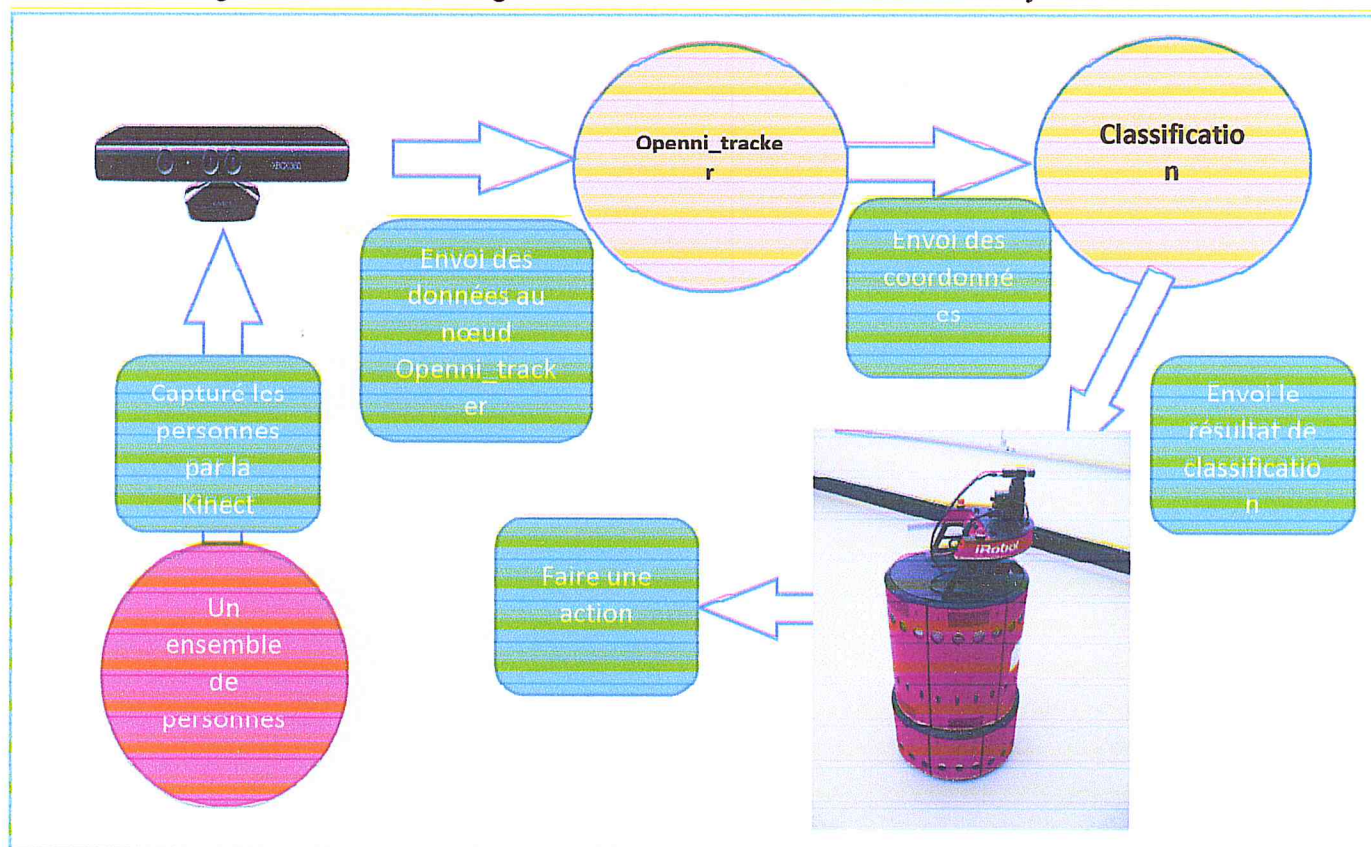


Figure 47 : architecture globale

3. Architecture de l'application :

La figure suivante illustre globalement le fonctionnement de notre système :



4. Environnement de Développement :

Le nœud de la classification a été développé dans le contexte de la plateforme ROS. Le langage utilisé est Python dans le cas du nœud récepteur 'classification' et le c++ pour le nœud émetteur 'OpenNI_tracker'.

La conception et réalisation du nœud ROS suit une démarche bien précise qu'il est nécessaire de maîtriser. La plus grande complexité que nous avons rencontrée c'est le fait que ROS utilise la notion de la programmation Event-Driven. Ainsi malgré la présence d'une méthode claire de mise en place de nouveau package pour ROS, la mise en place réelle d'un nouveau package est très délicate et nécessite un temps important pour la compréhension et la mise en œuvre. Dans ce qui suit nous allons illustrer comment un package peut être créé sous ROS. Par la suite nous exposerons, par les exemples les diverses fonctionnalités de notre approche.

5. Package ROS :

5.1. Définition d'un Package ROS :

C'est l'unité principale d'organisation logicielle de ROS. Un package est un répertoire qui contient les nœuds (nous verrons ci-dessous ce qu'est un nœud), les bibliothèques externes, des données, des fichiers de configuration et un fichier de configuration XML nommé manifest.xml.

5.2. Création d'un nouveau package sous ROS :

Pour créer un nouveau Package sous ROS on a suivi les étapes d'installation mentionnées dans le site⁵ officiel de ROS.

1. Il faut créer un environnement de travail.
2. on doit exécuter la commande fournie par ROS qui sert à créer un package :

```
catkin_create_pkg notre_package
```

3. La commande produit un dossier qui contient l'ensemble de fichiers nécessaires pour développer sous ROS :

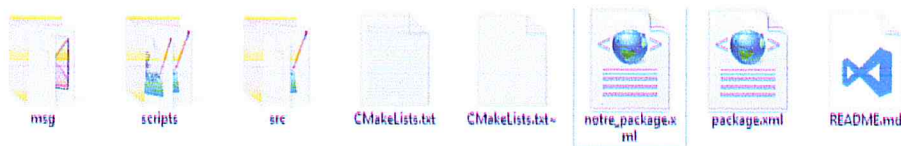


Figure 48 : les fichiers du package 'notre_package'

4. après la création du package il faut ajouter les dépendances, comme nous avons développé sous python et c++ il faut ajouter : rospy pour python et roscpp pour c++ dans le fichier XML notre_package.xml :

```
<package>  
<buildtool_depend>catkin</buildtool_depend>  
<build_depend>roscpp</build_depend>  
<build_depend>rospy</build_depend>  
</package>
```

5. à la fin il faut compiler l'environnement de travail.

⁵ <http://www.ros.org/>

6. Tests des fonctionnalités :

6.1. Tests effectués en simulation :

Pour éviter tous dégâts sur la plateforme robotique B21R, nous effectuons des tests de simulation sur l'ordinateur seulement.

Test pour une seule personne :

Les figures ci-dessous montrent le test de classification pour une seule personne.

Le rectangle en couleur verte représente la personne et sa position par rapport à la camera.

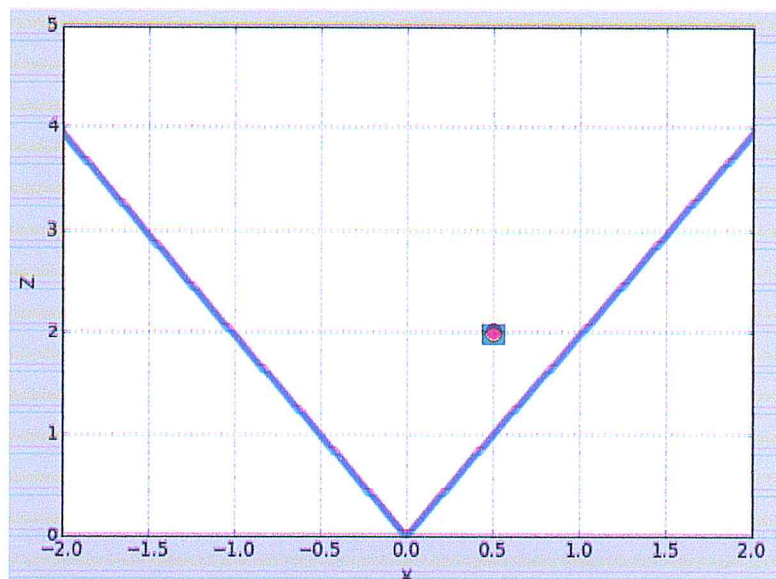


Figure 49 : teste de simulation pour une seule personne

L'affichage sur console indique que la liste finale contient une seule personne.

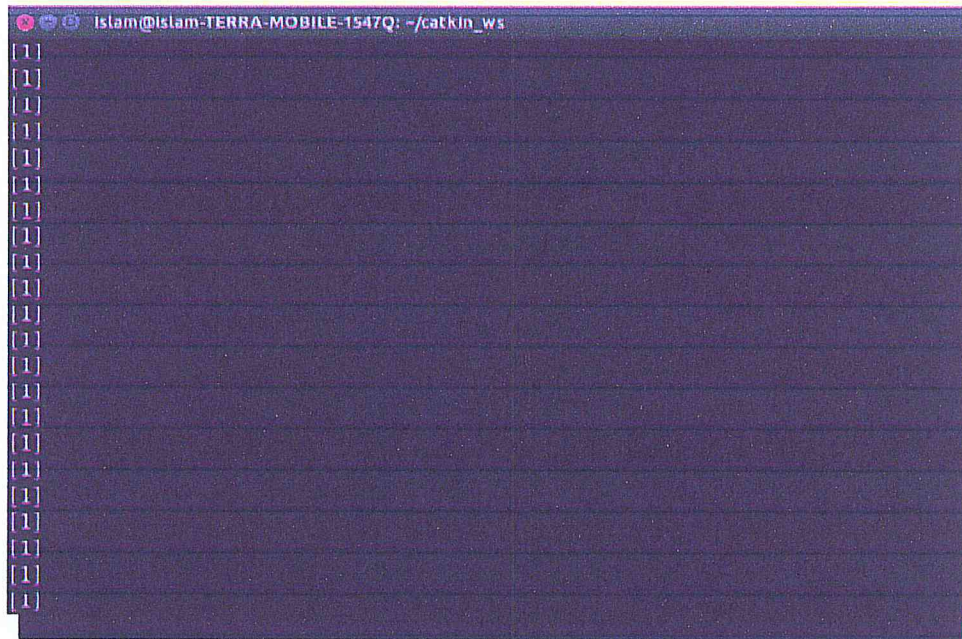


Figure 50 : résultat du premier test

Test pour deux personnes :

Les figures ci-dessous montrent le test de classification pour deux personnes.

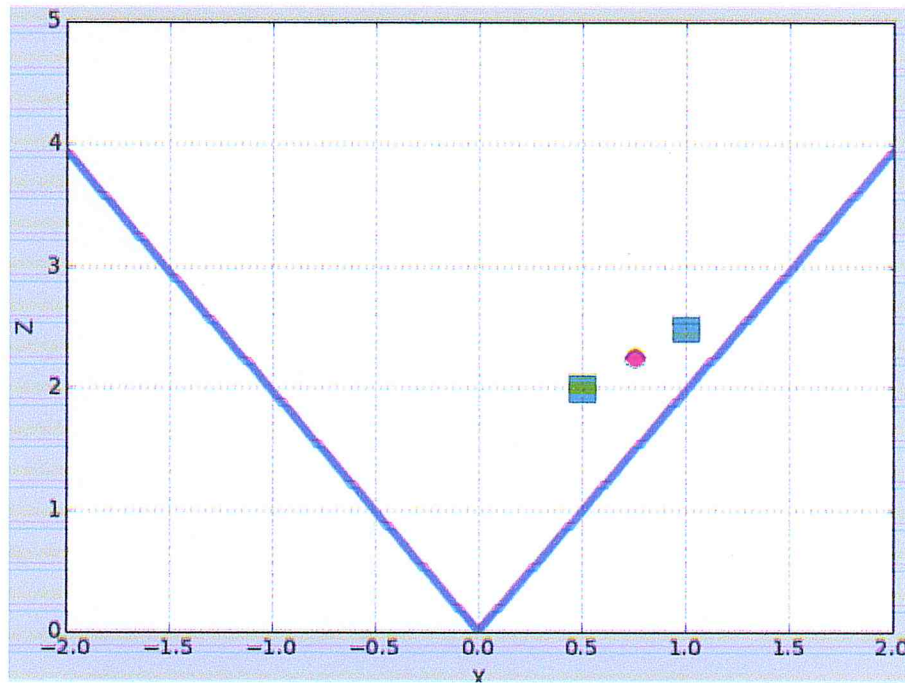
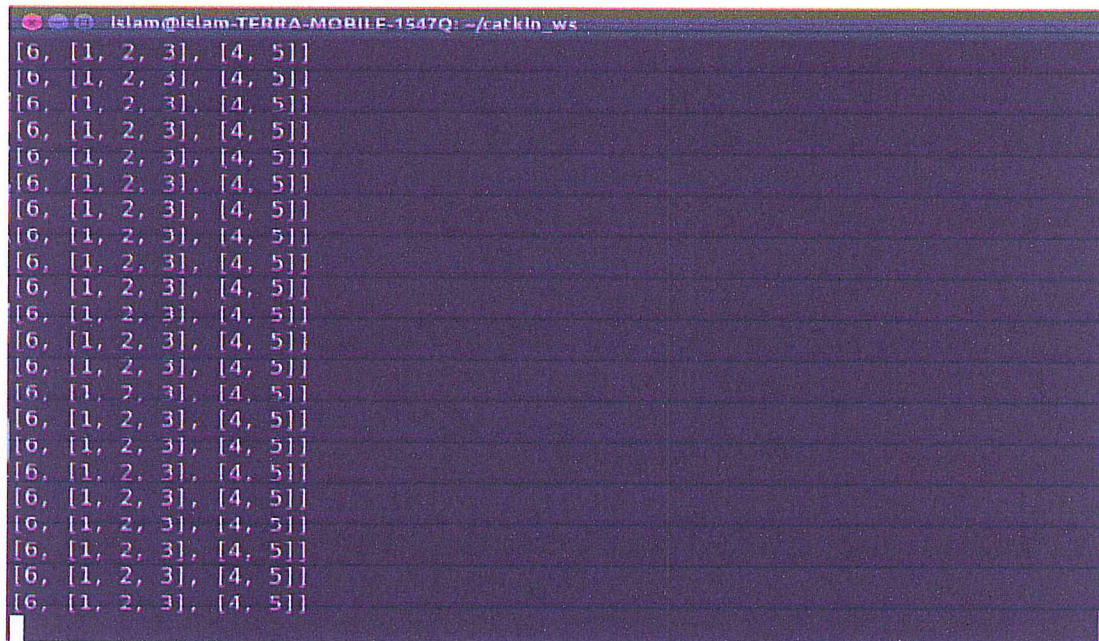


Figure 51 : test de simulation pour deux personnes

Le point en couleur rouge représente le centre de gravité du groupe.

Le point en couleur rouge représente le centre de gravité de chaque groupe, le point rouge dans le centre du rectangle vert représente le centre de gravité de la personne si elle est isolée.



```
Islam@Islam-TERRA-MOBILE-1547Q: ~/catkin_ws
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
[6, [1, 2, 3], [4, 5]]
```

Figure 56 : résultat pour plus trois personnes

L'affichage sur console nous donne une très bonne signification, la personne isolée représenté par identifiant '6' est la première dans la liste, vient par la suite le groupe de trois personnes, à la fin le groupe de deux personnes (ou couple isolée).

6.2. Tests effectués sur la plate-forme robotique expérimentale réelle :

Après le succès de la simulation on passe à l'étape de déploiement de notre approche sur le robot B21R.

La figure suivante montre une photo pris au CDTA pour tester l'efficacité de notre approche dans un environnement réel.



Figure 57 : test de l'algorithme dans un environnement réel

Le résultat montré dans la figure suivante indiquent que notre approche marche parfaitement dans un environnement réel.

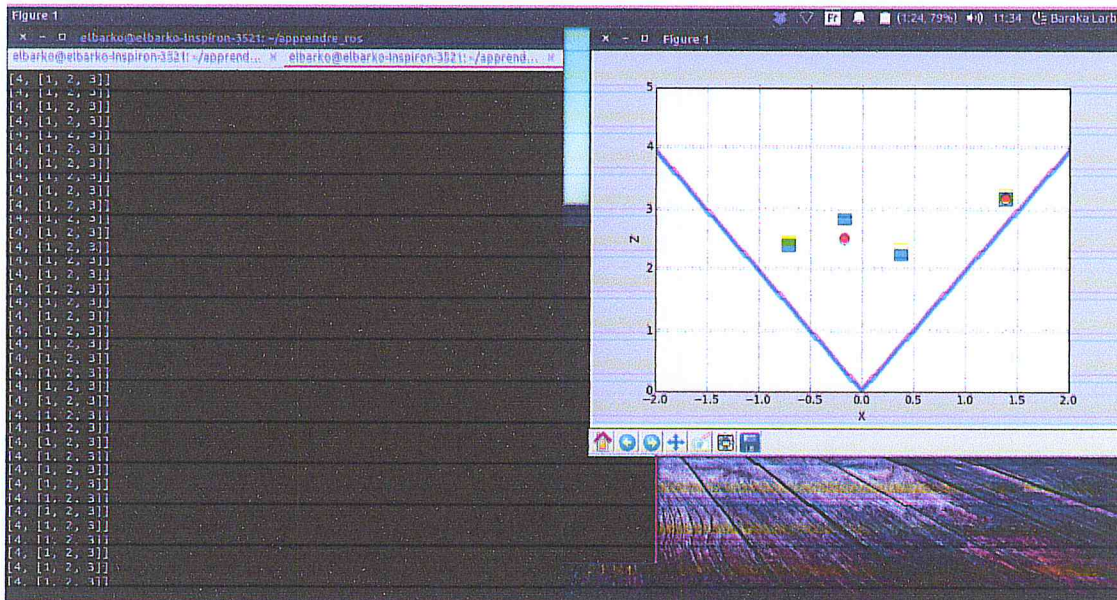


Figure 58 : le résultat du test

7. Performances :

Maintenant on procède aux tests de performances notamment le temps de réponse, pour cela nous nous sommes équipées d'un ordinateur qui possède les caractéristiques suivantes :

- Le Cpu : Intel i3 2.3 GHZ quadcore.
- Mémoire vive (RAM) : 4Go.
- Système d'exploitation : Ubuntu 14.04.

Pour cette configuration les composants exécutent en temps réel à une fréquence de 25-30hz, même dans un environnement encombré, quand il Ya plus de 30 personnes visibles.

Pour une fréquence a 30hz le temps de repense correspondant est 33ms, qui représente le seuil maximal limité pour notre approche.

La figure ci-dessous montre un graphe du temps de réponse par rapports aux nombres de personnes visibles

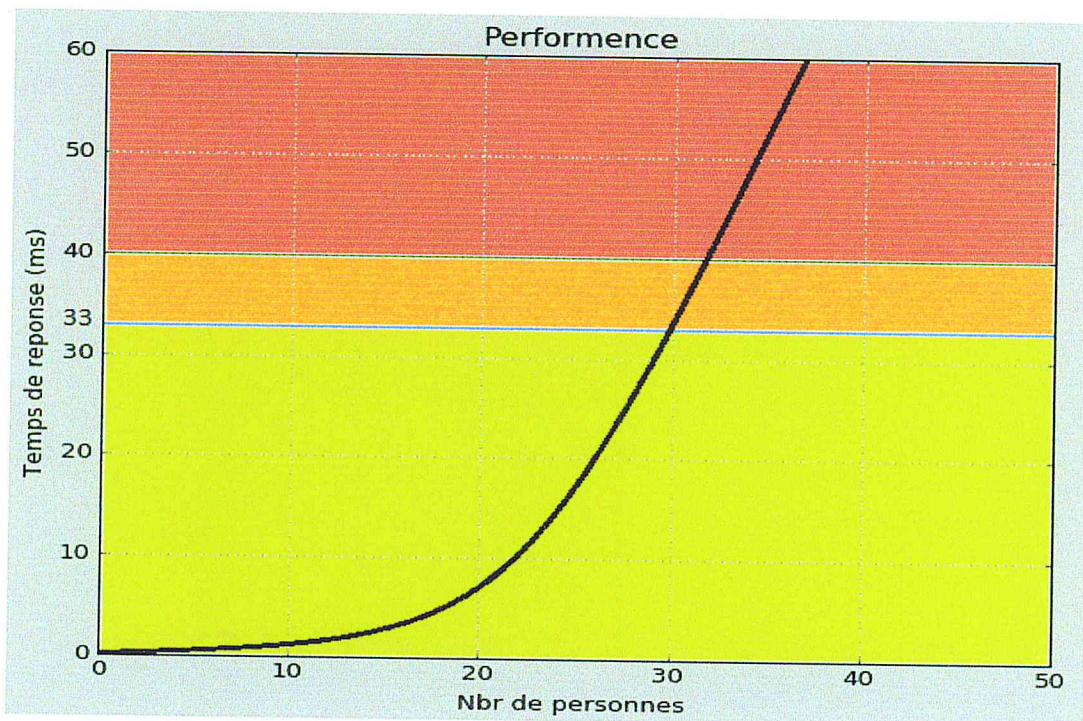


Figure 59 : Temps de réponse par rapport aux nombre de personnes

8. Fiabilité :

Dans notre cas la précision des résultats est primordiale où l'erreur est nulle, dans le choix de la classification hiérarchique ascendante. Il reste le problème de la grande complexité qui devient un facteur qui limite la mise en échelle pour un environnement très vaste.

9. Conclusion :

En conclusion, à travers ce chapitre nous avons démontré que notre approche a réussi à déterminer la classification des personnes que se soit une seule personne, un couple ou un groupe de personnes.

Cette expérimentation a démontré que c'était possible de donner une intelligence à cette détection c'est-à-dire, une intelligence au robot lui-même.

Conclusion générale :

Ce travail rentre dans le cadre de l'application des techniques de classification à l'implémentation des éléments (nœud) existant dans les différents équipements du robot.

Notre étude a porté principalement sur l'utilisation d'un équipement récent de détection et suivi de personnes qui est la Kinect et de faire une classification au vu de son utilisation pour les nœuds qui seront développés dans le futur.

A travers le travail accompli et les efforts octroyés tout au long de la réalisation de ce projet, nous avons la certitude d'avoir enrichi nos connaissances d'un professionnalisme que nous souhaitons utile et profitable dans le domaine de la vision par ordinateur.

Nous avons exercé dans notre mémoire l'étude des différentes approches de classification de l'être humain, dans le but de fournir un comportement social et une intelligence plus développée.

Dans le premier chapitre nous avons exposé Définition et Historique de la robotique, ainsi les types de robot pour chaque environnement après on s'est intéressé à la robotique de services en montrant les détails de cette dernière, après on a clôturé le chapitre par une définition du middleware robotique, enfin nous avons terminé par une brève description du système robotique ROS.

En ce qui concerne le deuxième chapitre, nous avons décrit La Détection d'objet et les approches de suivi ainsi la détection et le suivi en 3D, après nous avons exposé le capteur de profondeur « Kinect » et ses différentes caractéristiques, ensuite un résumé sur les différents travaux sur la détection et le suivi de l'être humain par les capteurs RGB-D.

Dans le troisième chapitre nous avons introduit la notion de classification en présentant quelques algorithmes existants.

Dans le quatrième chapitre, nous avons présenté la partie conception du projet en commençant par un schéma général de notre solution en détaillant chaque étape de traitement qui commence par une acquisition des données jusqu'à la classification en clarifiant par des diagrammes, images et des exemples.

Dans le cinquième et dernier chapitre, nous avons présenté l'architecture globale de notre système robotique et les différents tests appliqués sur la Kinect et sur la fiabilité du system.

Problèmes rencontrés :

Durant la conception du Project nous avons rencontré des soucis :

- Les erreurs liées à la détection lors du déplacement du robot
- Dans l'état de déplacement du robot n'est pas capable de suivre plus d'une personne à la fois.
- La position PSI est très gênante pour une rapide détection

Perspectives :

Dans la continuité de notre travail, nous devons traiter les propositions suivantes :

- Améliorer la visualisation 3D
- Introduire la reconnaissance vocale
- Implémentation d'une méthode d'un suivi d'un groupe de personnes

Bibliographie:

- [1] : Ernst Hoffmann, World robotics, par l'International Federation of Robotics, 2009.
- [2]: George Bekey, International Assessment of Research and development in Robotics, rapport pour la National Science Foundation (NSF) USA, 2006.
- [3]: iti Techmedia , Advanced Robotics, Market Intelligence Report, 2008.
- [4]: Gianmarco Veruggio, URON Roboethics Roadmap, 2007.
- [5]: Normann ANDERSEN, Robots on the Move from the Production Line to the Service Sector: The Grand Challenges for Contractors, Workers, and Management, 2010.
- [6]: Erwin Prassler ET Kazuhiro Kosuge, Domestic Robotics, Springer Handbook of Robotics, 2008.
- [7]: Global Industry Analysts, Service Robotics: A global Market Report, 2010.
- [8]: l'Association française de téléassistance, Le marché français de la téléassistance, présentation disponible sur le site de la CNSA, 2008.
- [9]: Computing Community Consortium, A roadmap for US Robotics: From Internet to Robotics, 2009.
- [10]: T.D.Grove, K.D.Baker, and T. N. TAN, Color based object tracking. 14th International Conference on Pattern Recognition (CV41).
- [11]: Paul Fieguth, Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates. 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97).
- [12] : W. Krattenthaler, K.J. Mayer et M. Zeiller, corrélation points : un modèle de coûts réduits correspondant Technique. *Proc ICIP*, 208 pp. 212, 1994.
- [13]: A.K. Jain, Y. Zhong et S. Lacroix, objet correspondant à l'aide de modèles déformables. *IEEE trans. Pattern Analysis and Machine Intelligence*, vol. 18, no 3, pp. 267-278, mars 1996.
- [14]: C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, Efficient and Effective Querying by Image Content. *J. Intelligent Information Systems*, vol. 3, no. 1, pp. 231-262, 1994.
- [15]: A. Makarov, Comparison of background extraction based intrusion detection algorithms, Dans *International Conference on Image Processing ICIP '96*, pages 521–524, 1996.

- [16]: Natan Peterfreund, Robust Tracking of Position and Velocity with Kalman Snakes. IEEE transactions on pattern analysis and machine intelligence, vol. 21, no. 6, June 1999.
- [17]: N. Dalal and B. Triggs.: Histograms of oriented gradients for human detection. CVPR, 1 (2005) 886-893.
- [18]: K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. CVPR 2(2004) 53-60.
- [19]: N. Dalal, B. Triggs, C. Schmid.: Human detection using oriented histograms of flow and appearance, in: European Conference on Computer Vision, Graz, Austria, May 7–13, 2006
- [20]: S. Ikemura, H. Fujiyoshi.: Real-Time Human Detection using Relational Depth Similarity Features. ACCV 2010, Lecture Notes in Computer Science, 2011, Volume 6495/2011, 25-38.
- [21]: A. LEJEUNE, S. PIÉRARD, M. VAN DROOGENBROECK et J. VERLY. Utilisation de la Kinect. Linux Magazine France, N°151, pages 16-29, Juillet-Août, 2012.
- [22]: Ni Bingbing, “Multi-Level Depth and Image Fusion for Human Activity Detection”, IEEE Trans. On Cyb, Vol 43, N° 5, October 2013, pp: 1383 – 1394.
- [23]: O.Serdar, « 3D Rigid Body Tracking Using Vision and Depth Sensors”, IEEE Trans. On Cyb, Vol 43, N° 5, October 2013, pp: 1953 – 1405.
- [24]: H.Zhang, « Real Time Multiple Human Perception with Color-Depth Camera on a mobile Robot », IEEE Trans. On Cyb., Vol 43, N° 5, October 2013, pp: 1429 – 1441.
- [25]: Kar, Abhishek. (2011). Skeletal Tracking using Microsoft Kinect. Department of Computer Science and Engineering, IIT Kanpur.
- [26]: Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In CVPR (1), pages 511{518, 2001}.
- [27]: Détection de l’homme en milieu intérieur pour le cas d’un robot guide, BOULARABI Brahim, Mme I. BOUDOUANE et Mme S. KAHLOUCHE, CDTA, 2014.
- [28] : Classification de la population en catégories socio-économiques : méthodologie et application pratique par Mustapha HADD Institut national de statistiques et d’économie appliquée - Ingénieur d’Etat Option : Démographie 1999
- [29]: A. Valero, G. Randelli, F. Botta, D. Rodríguez-Losada, M. Hernando « Operator Performance in Exploration Robotique » Journal of Intelligent & Systems, (1-21), 2011

[30] : Samuel Tardieu, Système d'exploitation embarqué, Telecom ParisTech , 2012

[31]: Johannes Maurer, ROS basic Concepts, Institute for software Technology, Austria, 2012

[32]: onathan Bohren, ROS Crash-Course, Part 1: Introduction to ROS distribution, build system and infrastructure, Johns Hopkins University, 2010.