

**UNIVERSITÉ PIERRE ET MARIE CURIE**  
**ÉCOLE NATIONALE SUPÉRIEURE DES TECHNIQUES AVANCÉES**

PROJET DE FIN D'ÉTUDES

---

**Méthode de reconnaissance d'objets  
multi-vues sur un robot mobile**

---

**Auteur**  
Luigi FRANCO TEDESCO  
*tedesco@ensta.fr*  
Promotion 2015

**Professeurs Responsables**  
David FILLIAT  
*david.filliat@ensta.fr*  
Safia KEDAD-SIDHOUM  
*Safia.Kedad-Sidhoum@lip6.fr*

**Tuteur**  
Jean-François GOUDOU  
*jean-francois.goudou@thalesgroup.com*

THALES Services | Campus Palaiseau  
828 Boulevard des Maréchaux, 91762 Palaiseau

Stage effectué du 09 mai 2015 au 28 août 2015



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Contexte . . . . .	6
1.2	Objectifs . . . . .	6
1.3	État de l'art . . . . .	6
<b>2</b>	<b>Méthode proposée</b>	<b>8</b>
2.1	Architecture générale . . . . .	8
2.2	Segmentation . . . . .	9
2.2.1	Algorithme . . . . .	10
2.2.2	Restrictions . . . . .	10
2.3	Descripteurs . . . . .	11
2.4	Reconnaissance mono-vue . . . . .	11
2.5	Localisation et suivi d'objet . . . . .	11
2.5.1	Définition de repères . . . . .	11
2.5.2	Transformation de repères . . . . .	12
2.5.3	les bases mobiles . . . . .	12
2.5.4	Filtre de Kalman . . . . .	12
2.6	Reconnaissance Multi-vue . . . . .	13
2.6.1	Chaînes de Markov Cachées . . . . .	13
2.6.2	Algorithme de Viterbi . . . . .	14
2.6.3	Graphe d'aspect polaire . . . . .	14
<b>3</b>	<b>Protocole Expérimental</b>	<b>16</b>
3.1	Matériel utilisé . . . . .	16
3.2	Setup expérimental . . . . .	16
3.3	Résultats expérimentaux . . . . .	17
3.3.1	Comparaison à la reconnaissance mono-vue . . . . .	17
3.3.2	Suivi et reconnaissance multi-cibles . . . . .	18
3.4	Problèmes rencontrés . . . . .	18
<b>4</b>	<b>Conclusion</b>	<b>20</b>
4.1	Synthèse . . . . .	20
4.2	Discussion . . . . .	20
4.3	Perspectives . . . . .	20
<b>A</b>	<b>Matériels</b>	<b>22</b>
A.1	Plateforme mobile . . . . .	22
A.2	Ordinateur Portable . . . . .	23
A.3	Capteur RGB-D . . . . .	23
<b>B</b>	<b>Logiciels</b>	<b>24</b>
B.0.1	Restrictions logiciels . . . . .	24

<b>C Base de données</b>	<b>25</b>
C.1 Segmentation . . . . .	26
C.1.1 Paramètres . . . . .	26
C.1.2 Floor Detection . . . . .	26
C.1.3 Estimation de la normale . . . . .	26
C.1.4 RANSAC . . . . .	26
C.2 Descripteurs . . . . .	27
C.2.1 Point Feature Histogram - PFH . . . . .	27
C.2.2 Fast Point Feature Histogram - FPFH . . . . .	27
C.2.3 Viewpoint Feature Histogram- VFH . . . . .	28
C.2.4 Clustered Viewpoint Feature Histogram - CVFH . . . . .	28
C.2.5 Déplacement du robot . . . . .	28
C.2.6 Fusion de données . . . . .	29
C.3 Planning de Travail . . . . .	29
<b>D Annexe II</b>	<b>30</b>
D.1 Le groupe Thales . . . . .	30
D.2 Secteurs d'activité . . . . .	30
D.3 Présentation de ThereSIS . . . . .	31
D.4 Secteurs d'activité . . . . .	31
D.5 Le laboratoire Video Technologies & New Sensors . . . . .	32

# Chapitre 1

## Introduction

EN VERT SONT DES PARTIES QUE NE SONT PAS ENCORE FINI

### 1.1 Contexte

La perception de l'environnement par des machines est indispensable pour leur intégration à la vie quotidienne. Des compétences telle que se localiser, la prise de décisions et une capacité d'apprentissage sont nécessaires, même pour la réalisation des tâches les plus simples. Dans cet étude, on s'intéresse à la compréhension d'éléments constituant une scène, sujet récurrent dans le domaine de la vision par ordinateur, et à l'apport de l'utilisation d'un robot mobile dans cette tâche. *Ici, tu peux peut être parler de la représentation des objets dans le cerveau avec des vues et des transitions entre vues et dire qu'il peut être intéressant d'explorer cette approche plutôt que de faire de la reconnaissance statique, image par image.* Plusieurs approches proposés dans la littérature explorent une sous-partie du *pipeline* de la reconnaissance pour faire face aux difficile défi de représenter les caractéristiques visuelle des objets *citation ?*, pendant que d'autres s'intéressent à l'utilisation du système de reconnaissance pour des tâches de recherche dans l'environnement et la manipulation d'objets *citation ?*.

### 1.2 Objectifs

Notre démarche correspond, initialement, à l'intégration de techniques de l'état de l'art pour arriver à un système fonctionnel de reconnaissance intégré sur une plateforme mobile équipée d'un capteur RGB-D. Dans un premier temps, la plateforme doit être capable d'acquérir une base de données d'images d'objets de manière automatisée. Ensuite, le but est d'utiliser ces informations apprises pour vérifier si un objet candidat est ou non celui présenté auparavant. Finalement, nous souhaitons utiliser les information de son déplacement pour renforcer sa perception, lever les possibles ambiguïtés et rendre le système moins sensible à différents types de bruits. De plus, tous ces algorithmes doivent fonctionner en temps réel et être implémentés pour fonctionner sur le robot présent au sein du laboratoire.

### 1.3 État de l'art

*Si tu as des papiers qui sont des revues de litteratures, tu peux les citer au début*

La majorité de la littérature traite le problème de la reconnaissance d'objets basés sur une seule image. Typiquement, un ensemble de *features* [9] *mettre des exemples de features en citations* est extrait et ensuite comparé aux modèles d'objets présents dans une base de données initiale. Il existe également des méthodes directes, comme deep learning *mettre une citation*, où l'image d'entré est

associée directement avec des classes des objets correspondants, au prix d'une étape d'entraînement importante d'apprentissage, encore plus dans un espace à 3 dimensions provenant du capteur RGB-D **bizarre cette fin de phrase ....** De très nombreux exemples d'applications de la reconnaissance mono-vue existent dans le domaine de la robotique, pour la navigation sémantique [6], couplé avec l'estimation de pose pour la saisie de l'objet [5] ou encore pour la recherche d'objets dans l'environnement [10, 1].

Des effort conséquent ont été mis en oeuvre fait pour améliorer l'extraction, le *matching*, ainsi que les *features* elles-mêmes pour qu'elles soient invariantes à transformations affines de l'image et plus représentatives de l'objet **cite des exemples de features 2d (A-SIFT, Harris affine) et 3d (pfh ...)** **qui font ça. Tu peux peut être mettre une citation ou deux sur le tracking (2d et 3d) en disant que ca aide à être plus robuste au cours du temps.** Ce traitement classique a l'avantage d'être à la fois modulaire, avec des étapes bien définies de segmentation, d'extraction de features, de classification et de post-traitement, et en même temps, d'avoir des résultats satisfaisants sur des cas simples.

Malgré l'intérêt des features invariantes, on s'aperçoit rapidement de leurs limitations lorsque des vues ambiguës apparaissent. Un premier travail s'inspire de la continuité, temporalité et séquentialité des observations dans la reconnaissance chez les humains pour augmenter la représentativité des modèles d'objet, et ainsi surmonter la limitation de la reconnaissance mono-vue **citation.** Certaines approches se basent sur des modèles CAD à trois dimension, la description de contours et les graphes [12] pour augmenter leur représentativité. En particulier, les graphes d'aspect **citation** permettent une représentation basée sur une composition d'images de points de vue différents et les liens entre elles. Les vues représentatives, nommés *key-Frames*, peuvent être choisie avec des politiques aléatoire, constant ou à la recherche d'événements visuels **des citations ! tu devrais aussi parler du travail de cedric [11], de l'article que jeff t'avait donné sur les radars, et tu peux aussi citer un mec de l'ensta qui a fait de la reconnaissance temporelle basée sur des graphes (vérifier si c'est bien celui la) [7]**

Enfin, certaines approches vont encore plus loin en suggérant une reconnaissance active. Ici, une estimation de quel serait le meilleur déplacement pour lever des ambiguïtés permet de repérer le capteur. Cela peut se faire par des critères de réduction de l'entropie en utilisant des probabilités de reconnaissance antérieures [3], en utilisant l'apprentissage par renforcement [2] où encore par estimation des faces cachées de l'objet **je n'ai pas trouvé la citation** ref :*Next Best View*. Finalement, [4] traite ce même problème comme un problème de localisation et suivi par un filtre particulaire.

L'utilisation d'un algorithme de reconnaissance basé sur une seule image possède l'inconvénient de ne pas prendre en compte les notions de vue et de transition entre elles. Pourtant, la majorité de ces systèmes souhaitent être invariants aux différents points de vue des objets, en d'autres termes, avoir la capacité de l'identifier de n'importe quel point de vue. Un système mono-vue pourrait traiter le concept de vues les plus représentatives et des transitions, mais cela de façon moins intuitive. Ainsi certains des articles présentés à la section ?? travaillent sur le domaine multi-vues en y intégrant des aspects géométriques et ainsi augmenter la qualité de son estimation. **Ce paragraphe aurait sa place dans l'état de l'art non ?**

# Chapitre 2

## Méthode proposée

L'objectif de notre méthode est d'avoir une reconnaissance multi-vues d'un ou plusieurs objets à la fois, capable d'intégrer le déplacement du robot pour résoudre des ambiguïtés et faux positifs. Pour incorporer les notions de vues et de transition entre elles, on utilise une représentation simple et suffisamment générale basée sur les graphes d'aspect. Le déplacement d'un état à un autre dans ce graphe est ensuite estimé par rapport au déplacement du robot. Ce système est ensuite couplé avec un dispositif de reconnaissance mono-vue classique capable de retrouver la vue la plus probable d'un objet à partir de descripteurs 3D. Une méthode de suivi des objets et un traitement probabiliste de changement de vue étant donné l'information motrice permet enfin d'augmenter le taux de reconnaissance.

### 2.1 Architecture générale

L'approche a été développée pour une base mobile différentielle munie de capteurs proprioceptifs odométriques et d'une caméra RGB-D. Les informations provenant des ces unités sont envoyées à une unité de traitement qui interprète les images reçues, isole les objets qu'elles contiennent et compare cette interprétation avec une base de données stockée dans la mémoire. En cas d'absence de correspondant dans la base, ce nouvel exemplaire pourra éventuellement être ajouté à la base de données et agrandir les connaissances d'objets existants dans l'environnement.

L'architecture du système est illustrée à la figure 2.1 et permet à la fois, de comprendre les dépendances entre les étapes de traitement, de même que, la nature du flux d'information entre modules.

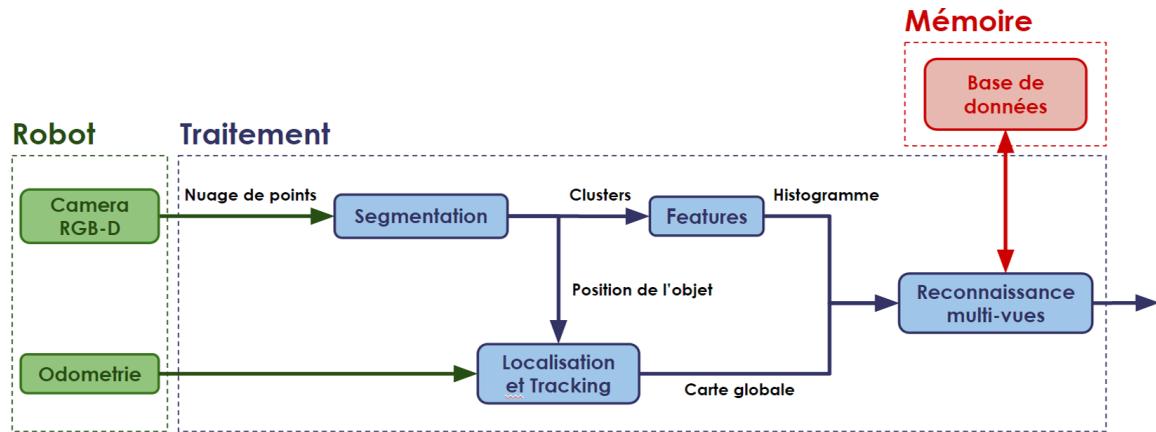


FIGURE 2.1 – Architecture générale du système

Plus précisément, l'unité de traitement reçoit un nuage de points brut provenant de la caméra, ainsi que la mesure de rotation de roues du robot. La première partie du traitement vise à nettoyer le nuage en segmentant des points des objets candidats de la scène, ce qui permet d'enlever une partie non pertinente et de créer un nuage de point dédié pour chaque objet de l'image. Ces nuages de points sont envoyés à l'unité d'extraction de features pour générer des histogrammes représentatif de chaque objets dans l'image. Simultanément, une conversion de référentiel localise des objets dans le repère absolu de déplacement du robot. Puis, les positions des objets sont données au module de localisation et tracking qui suit les observations et les associe entre elles pour avoir une cohérence globale des positions. En dernier lieu, pour chaque objet, les histogrammes de features ainsi que leurs positions converties dans un repère global sont envoyées au module de reconnaissance, et sont utilisés pour reconnaître les éléments de la scène et donner leur vue la plus probable à chaque instant de temps.

Les prochaines sections détaillent l'architecture présentée à l'image 2.1, en présentant le fond théorique derrière le fonctionnement de chaque sous-module.

## 2.2 Segmentation

La segmentation consiste à isoler des objets dans une image brute, ou en d'autres termes, différencier les éléments qui ne constituent pas un objet et les objets eux-mêmes. La segmentation d'objets est considérée comme un élément essentiel en traitement d'image étant donné qu'une fois l'objet séparé du fond, la reconnaissance devient beaucoup plus simple. La difficulté majeure d'un tel algorithme sur des images RGB vient du fait que la projection de la scène sur le plan image supprime l'information de profondeur. Les capteurs stéréoscopiques et infra-rouges permettent de compenser cette absence d'information et simplifient énormément le traitement nécessaire. **La dernière phrase n'est pas forcément utile je trouve.** Les cartes de profondeur pourraient être utilisées pour représenter l'information de profondeur, pourtant, encore plus naturel, le concept de nuage de points propose une représentation en trois dimensions de l'environnement capturé.

Je t'ai ajouté une phrase pour la segmentation dans le cas où le fond est statique. Dans le cas où le capteur est immobile, on utilise classiquement des méthodes de soustraction de fond pour l'étape de segmentation [? ]. Ceci n'est pas possible dans notre cas car le robot évolue dans son environnement. La démarche proposée par la littérature dans ce cas considère les objets comme des ensembles de points délimités par un seuil de proximité. Cette définition est suffisamment générale pour permettre de représenter une grande quantité d'objets. Néanmoins, définir ces ensembles dans une image brute n'est pas forcément simple. Par conséquent, on utilise un nouvel *a priori* qui spécifie que les objets se situent sur des plans de support. Bien que plus restrictif que la définition

d'avant, cela permet un segmentation crédible. Parmis les méthodes de segmentation se basant sur cette définition, on peut citer Tabletop object detector [?] qui détermine le plan principal de l'image (généralement une table ou le sol) grâce à l'algorithme RANSAC [8], puis recherche des objets dans l'enveloppe convexe de ce plan. Par ailleurs, Caron et al. [6] ont proposé une approche légèrement différente. En partant du même principe, le sol est estimé, puis un traitement pour le fond de la scène est appliqué, où les plans orthogonaux à la normale du sol et de taille suffisamment grands sont considérés comme des murs, et les éléments trop près des bords ne sont pas considérés.

### 2.2.1 Algorithmes

La méthode de segmentation utilisée dans notre cas est celle proposée par Caron et al. Cette méthode s'applique surtout pour de la segmentation d'objets posés sur le sol dans des environnements intérieurs et répond aux exigences du domaine de déplacement du robot : le laboratoire de Thales Therésis.

Plus spécifiquement, elle peut être découpée selon les étapes suivantes :

0. Calibration permettant d'obtenir l'équation du sol avant le début de la séquence. Puis, pour chaque frame de la séquence
1. Soustraction du sol à partir de l'équation trouvée
2. Filtrage des points trop éloignés, considérés comme plus incertains
3. Calcul de la normale des surfaces de la scène
4. Élimination de murs, considérés comme des plans orthogonaux au sol et de taille suffisamment grande
5. Voxelisation des points non filtrés pour accélérer le traitement
6. Projection des points voxelisés dans le plan du sol
7. Regroupement des points en objets grâce à l'algorithme de *clustering point growing* de PCL
8. Calcul du centroïde et des bounding boxes 2D et 3D de chaque objet

Ainsi, l'algorithme fournit la position de chaque objet dans le repère de la caméra ainsi que le nuage de point et les normales qui leur sont associées.

Une calibration initiale est nécessaire pour définir l'équation du sol. Pour cela, on place le robot dans un endroit où l'image obtenue correspond majoritairement au sol. L'équation du plan dominant est extrait par RANSAC et sauvegardée dans un fichier texte. **Ce n'est peut être pas la peine de détailler trop en annexes. Tous les algos sont classiques, il suffit de mettre une référence ou un lien vers l'implémentaiton utilisée.** Une explication plus détaillée sur les sous-méthodes utilisées pour chaque étape est présentée dans les annexes, ainsi qu'une discussion sur les paramètres utilisés.

### 2.2.2 Restrictions

**Ce paragraphe aurait peut être plus sa place dans les annexes. Vois en fonction de la place qu'il te reste ...** La physique des capteurs restreint le type d'objets qui peuvent être aperçus et segmentés, soit à cause des réflexions des rayons infra-rouges, soit à cause de la résolution limitée des images mesurées. D'un autre côté, la segmentation a ses propres contraintes en ce qui concerne le positionnement des objets dans l'image et, principalement, la définition du sol et des murs. Par conséquent, les restrictions de l'algorithme sont les suivantes :

- L'objet se trouve par terre.
- L'objet se trouve au centre de l'image.
- La lumière ambiante ne doit pas contenir trop de lumière infra-rouge.
- L'objet n'est ni transparent ni trop réfléctif.
- Le sol où le robot se déplace n'est pas accidenté.
- L'objet se trouve à une distance inférieure à 3 mètres.
- L'objet est assez grand et dépasse le seuil d'appartenance au sol.

Un grand nombre d'objets, entre autres chaises, tables, écrans, boîtes en carton, poubelles, de tailles et formes variés ont été testés et peuvent être segmentés malgré les restrictions. Quelques exemples de segmentation sont présentés en annexes **il est mieux de citer l'annexe à laquelle tu fais référence. Dans ton cas l'annexe ?? pour illustrer la capacité de segmentation. Si tu as de la place, tu peux carrément montrer des exemples dans cette section et non dans les annexes.**

## 2.3 Descripteurs

Le travail des descripteurs est, d'une part, d'extraire des caractéristiques intéressantes de l'élément observé et, d'autre part, de réduire la dimensionnalité de l'espace traité, tout en restant robuste à des transformations affines et aux changements de luminosité. On s'intéresse surtout ici aux descripteurs basés sur le nuage de point des objets, bien qu'il soit possible aussi d'utiliser des descripteurs associés à la texture ou à la couleur. Les descripteurs qui nous intéressent sont des descripteurs géométriques qui essaient de traduire les idées de courbure, de forme et taille dans les histogrammes, et sont intéressants pour étudier les ambiguïtés de reconnaissance. Parmis les descripteurs 3D proposés dans la littérature, on peut citer PFH [] qui est invariant par changement de point de vue. **décris les autres méthodes en une phrase avec la citation associée** Une description détaillée de ces descripteurs et leurs principales différences sont expliquées dans les annexes ???. Nous choisissons d'utiliser le descripteur *Viewpoint Feature Histogram - VFH*, car il permet de discriminer non seulement les formes géométriques (pour la reconnaissance d'objet), mais aussi les points de vues (reconnaissance de vue).

En partant de l'hypothèse que la segmentation propose un découpage correct des objets, on extrait des descripteurs globaux à partir des ensembles de points proposés. Ainsi, pour chaque objet, on obtient un histogramme VFH représentatif de l'objet et de la vue segmentée.

## 2.4 Reconnaissance mono-vue

matching des descripteurs avec la base de données...

L'étape de classification correspond à la différentiation entre les histogrammes caractéristiques de chaque vue de chaque objet. Cette mesure pourrait être apprise, par exemple, avec un réseau de neurone ou n'importe quel autre méthode classique de *machine learning*. **Le travail \* three dimensional dof cluster vfh...\* suggère l'utilisation de la mesure chi-squared similitude entre histogrammes accouplé au classificateur k plus proches voisins.** Le grand avantage de ce classificateur c'est l'étape d'apprentissage correspond à création d'un arbre de recherche, construit d'après la comparaison croisée entre les éléments de la base, que pour l'ordre de grandeur de la base de données envisagé, est presque instantané. Distance plus naturel...

L'API de la librairie FLANN sur PCL permet l'utilisation directe du classificateur K - plus proches voisins. L'implémentation permet l'utilisation de plusieurs définitions de distance entre histogrammes. La définition par défaut, Chi-squared, dont la formule est décrit dans la suite, semble être capable de bien différencier les histogrammes d'entrées,  $H_1$  et  $H_2$ , et était choisi comme la définition pour le classificateur.

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

## 2.5 Localisation et suivi d'objet

### 2.5.1 Définition de repères

Se placer dans différents repères permettre d'avoir de référentiels plus naturels pour chaque type de composant du robot et pour les objets placés dans la scène. On défini quelques repères et

conventions de base pour faciliter la localisation. D'abord le repère de la base du robot est orthonormale positive, où le déplacement vers l'avant correspond à l'axe  $x$ , vers la gauche à l'axe  $y$  et vers le haut à l'axe  $z$ . Une deuxième référentielle de convention égale à celle d'avant positionne le capteur RGB-D par rapport au robot. Enfin, le dernier référentiel correspond au repère optique du capteur orienté selon la convention usuelle pour les images avec l'axe  $x$  orienté vers la droite et l'axe  $y$  vers le bas et, enfin, l'axe  $z$  vers l'avant. Ces trois repères permettent d'orienter tous les éléments aperçus par le robot dans l'environnement de façon pratique.

La [figure](#) illustre ces repères utilisés et les conventions décrites.

### 2.5.2 Transformation de repères

Ensuite, la transformation entre repères permet la passage de l'un à l'autre, pour avoir la position globale de l'objet d'après sa détection par la caméra. La transformation entre une base  $a$  et une autre  $b$  est faite par une matrice de rotation et translation classique, décrite en bas.

$$\mathbf{R}_b^a = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta x \\ \sin \theta & \cos \theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$$

où  $\theta$  équivaut à la rotation entre les deux repères et  $\Delta x$  et  $\Delta y$  sont les translations linéaires entre eux.

### 2.5.3 les bases mobiles

#### Estimation de l'odométrie

Certains robots sont dotés de capteurs à estimer de façon approximée leur déplacement. C'est aussi le cas du robot ciblé qui possède encodeurs capables d'estimer la rotation angulaire des roues. Une intégration, au sens mathématique, de la différence entre l'odométrie entre deux intervalles de temps permet de retrouver la position globale du robot.

$$\begin{aligned} x_t &= x_{t-1} + \delta x_{t-1} * \cos(\theta_{t-1}) - \delta y_{t-1} * \sin(\theta_{t-1}) \\ y_t &= y_{t-1} + \delta x_{t-1} * \sin(\theta_{t-1}) + \delta y_{t-1} * \cos(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} + \delta \theta_{t-1} \end{aligned}$$

Les plates-formes différentielles Une des configurations les plus utilisées pour les robots mobiles d'intérieur est la configuration différentielle qui comporte deux roues commandées indépendamment. Une ou plusieurs roues folles sont ajoutées à l'avant ou à l'arrière du robot pour assurer sa stabilité (Figure 4.1). Cette plate-forme est très simple à commander, puisqu'il suffit de spécifier les vitesses des deux roues, et permet de plus au robot de tourner sur place. Cette possibilité permet de traiter dans certains cas le robot comme un robot holonome, ce qui va simplifier la planification de déplacement et la commande du robot.

L'estimation du déplacement par odométrie est également très simple à partir de la mesure des vitesses de rotation des deux roues et . Les vitesses de translation  $v$  et de rotation sont en effet données par :

Ce type de plate-forme peut également être utilisé avec des chenilles ce qui fournit une capacité de franchissement de petits obstacles intéressante (Figure 4.1). Ces plates-formes peuvent ainsi être utilisées en extérieur ou dans des décombres. L'utilisation de chenilles conduit cependant à une odométrie très bruitée à cause du contact mal défini entre les chenilles et le sol qui glissent beaucoup, notamment lors des rotations. L'estimation de la direction par l'odométrie sur ce type de plates-formes est donc en général rapidement inutilisable.

### 2.5.4 Filtre de Kalman

La modélisation des objets entraîne le besoin initial de les localiser dans la scène pour, postérieurement, les identifier. À cause de la divergence de l'odométrie, la mauvaise segmentation et le calcul du cen-

troïde de l'objet, la position estimée est fortement bruitée ayant un écart type qui rend la suive et identification infaisable lorsque plusieurs objets sont minimalement proches. Un filtre de Kalman ayant un modèle unitaire pour la matrice de transition d'états, moyenne les observations pour s'adapter au bruit de mesure.

Cependant, le caractère monomodal du filtre de Kalman fait en sorte qu'un seul objet cible peut être suivis à la fois. Pour atteindre l'aspect multimodal, il faut que plusieurs filtres tournent en parallèle. Ainsi, le problème passe d'estimer la position à décider quelle observation appartient à quel filtre, l'étape d'identification. Cela se fait à l'aide d'une matrice de corrélation de distances entre les nouvelles observations et les états courants de chaque filtre existant. Une solution simplificatrice est d'associer chaque observation au filtre selon l'ordre de vraisemblance de cette matrice. Lorsqu'une ambiguïté se produit dans l'étape d'identification, la classification peut aider à prendre une décision de mettre un filtre à jour ou, alors, créer un nouveau filtre.

Classiquement le filtre de Kalman est mis à jour dans deux étapes :

### Prédiction

Une première de prédiction que s'utilise du modèle mathématique décrivent l'évolution du processus

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k\end{aligned}$$

### Innovation

et une deuxième de mise à jour, où l'observation est incorporé pour le calcul de l'innovation e du gain de Kalman.

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}\end{aligned}$$

## 2.6 Reconnaissance Multi-vue

### 2.6.1 Chaînes de Markov Cachées

Le déplacement physique du robot résulte dans une séquence d'observations, en angles différents, d'un même objet. On exploite l'information odométrique entre les visualisations pour prédire les prochaines possibles orientations. De cette manière, l'évolution de la reconnaissance au long du temps est représenté par un processus stochastique, dont une modélisation possible correspond à le traiter de façon discrète dans un espace d'état. Ayant l'apriori que la dernière image et le dernier déplacement suffisent pour faire cette prédiction, en respectant, donc, la propriété de Markov de premier ordre, le processus stochastique est modélisée sur le cadre d'une chaîne de Markov cachée.

Concrètement, les états cachées correspondent à des objets connus au préalable et déjà incorporés dans la mémoire du robot. Cela constraint le nombre d'états et on se rencontre avec un chaîne fini. Puis, une matrice de transition,  $a_{i,j}$ , décrit l'évolution du processus et c'est là où l'odométrie et la relation entre vues et entre objets sont incorporés. Finalement, une autre matrice,  $P(y_1 | k)$ , dit matrice d'émission, estime la vraisemblance entre l'observation et les états de la chaînes.

Une autre deuxième modélisation serait d'avoir une chaîne de Markov Cachée distincte pour chaque objet et ensuite décider quel était le processus le plus vraisemblable. Ce cas est un sous-ensemble du cas antérieur où les transitions entre deux objets ne sont pas considérés. Pourtant, ce

qui peut être utile s'on considère l'évolution d'objets, par exemple, la transition entre une chaise vide et une personne assise sur une chaise ou encore un personne commence à marcher<sup>1</sup>.

### 2.6.2 Algorithme de Viterbi

Il reste, donc, extraire des informations de la modélisation Markovienne proposée. La séquence d'états la plus vraisemblable qui pourraient avoir générée les observations  $y_1, \dots, y_T$ , correspondrait exactement à la séquence d'objets reconnus. A fin de retrouver cette séquence, aussi appelé chemin, on fait appel à la programmation dynamique, spécifiquement à l'algorithme de Viterbi, d'où viens le nom chemin de Viterbi. L'algorithme retrouve de façon récursive l'état current le plus probable, prenant en compte seulement les observations jusqu'au instant donné et son estimation au instant intérieur, comme décrit par les équations suivants :

$$\begin{aligned} V_{1,k} &= P(y_1 | k) \cdot \pi_k \\ V_{t,k} &= \max_{x \in S} (P(y_t | k) \cdot a_{x,k} \cdot V_{t-1,x}) \end{aligned}$$

La probabilité que la séquence d'états le plus probable finissant dans l'état  $k$ , avait générée les observation au moment  $t$ , est sauvegardé dans  $V_{t,k}$ , pendent que  $\pi_i$  c'est la probabilité initiale de se rencontrer en chaque état. Pour retrouver le chemin de Viterbi, il suffit de trouver le maximum de  $V_{t,k}$  :

$$x_T = \arg \max_{x \in S} (V_{T,x})$$

### 2.6.3 Graphe d'aspect polaire

On considère que les objets sont décrits par deux dimensions d'information : une spatiale, concernant la position absolu de l'objet dans l'environnement et les positions relatifs où l'objet était visualisé, et une autre visuelle, donnée par les descripteurs géométriques, de couleurs et de texture ; qu'on cherche à transporter dans un référentiel unique. Le graphe d'aspect permet de coupler l'ensemble d'images suivant ses possibles transitions spatiales ce qui résulte dans la possibilité de construire le modèle à la volée et de jouer avec sa densité d'information - nombre d'images incorporées.

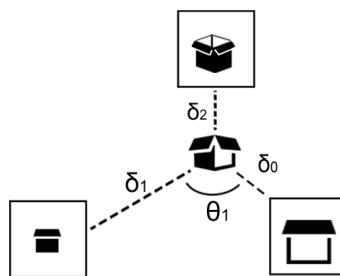


FIGURE 2.2 – Modèle polaire des objets.

Formellement, un référentiel polaire entrelace toutes ces informations de façon à représenter la position spatiale d'où l'observation était fait, tel comme il est représenté dans l'image <sup>\*7\*</sup>. Pour la construction du modèle les conventions suivantes étaient adoptées :

- l'angle zéro est attribué à la première observation
- L'origine du référentiel est la position globale de l'objet

<sup>1</sup>. Le fait de se mettre en mouvement altère les formes d'une personne, ce qui possibilite sa détection comme un nouveau objet.

- Les features sont labellisées d'après le déplacement angulaire et la distance au centroïde de l'objet.

Une grande majorité de features visuelles sont variantes à échelle, une fois que la résolution de l'image joue un rôle assez critique pour la détection de features, comme les patches SIFTs. Ainsi, avoir la distance que l'image étais prise peut être intéressant pour limiter la classification à une échelle valable.

# Chapitre 3

## Protocole Expérimental

### 3.1 Matériel utilisé

En ce que concerne les aspects matériels, le robot bimoteur Wifibot v2, équipé d'un ordinateur à bord, sera utilisé comme plateforme mobile. L'acquisition des données est faite par une caméra RGB-D Asus Xtion Pro Live. Par rapport au choix logiciel, l'environnement robotique ROS<sup>1</sup> a été adopté pour avoir les bibliothèques qui permettent de gérer les nuages de points, Freenect, OpenNi2 et PCL<sup>2</sup>, et d'autres nombreux outils de contrôle du robot et sauvegarde d'informations.

### 3.2 Setup expérimental

Pour évaluer les capacités de reconnaissance du robot, vingt objets de tailles et formes diverses ont été choisis pour être incorporé à la base de connaissance. Ils s'agit d'objets typiques qui peuvent être facilement retrouvés dans un laboratoire ou un bureau. Ensuite, nous avons effectué un tour complet de l'objet avec le robot en sauvegardant les nuage de point et en extrayant les features VFH pour huit positions différentes écartées de 45 degrés à 1.5 mètres de distance. La position correspondant à l'angle zéro a été choisie de manière aléatoire en alignant un des axes de l'objet avec celui du capteur. L'image 3.1 des vues d'un objet exemplifie la composition de la base de données. Une liste complète des objets figure en annexe ??.



FIGURE 3.1 – Les huit point de vues de l'objet commençant par la position zéro et en tournant le robot en sens horaire

- 
1. Robot Operating System
  2. Point Cloud Library

### 3.3 Résultats expérimentaux

#### 3.3.1 Comparaison à la reconnaissance mono-vue

Une première évaluation consiste à faire un tour complet autour de l'objet à reconnaître pour quatre positions angulaires différentes : 0, 45, 90 et une dernière choisie de manière aléatoire pour chaque objet. Le robot fait le tour à une vitesse de  $0.35 \pm 0.1m/s$  à une distance de  $1.5m$ , en enregistrant des images à  $1hz$ , ainsi, une expérience typique est constituée d'environ 25 images d'angle différent et prendre  $25 \pm 3$  seconds.

La difficulté de l'évaluation vient, premièrement, du fait du robot avoir une base assez discrète ce qui donne marge à des mauvaise reconnaissance mono-vue une fois le point de vue étant inexistant dans la base de connaissance. Puis, la vitesse du déplacement résulte dans images plus floues lors des acquisitions et la modification de l'angle de la caméra<sup>3</sup> apportent un obstacle en plus pour le *matching* de descripteurs dans le classificateur K-plus proches voisins.

Un expérience typique est illustrée dans l'image 3.2 :



FIGURE 3.2 – **Résultat de l'évaluation** - Reconnaissance multi-vue corrige des ambiguïtés et surmonte la mauvaise segmentation lors de la création de la base.

La première ligne correspond à la séquence d'images vues par le robot à chaque instant de temps, et donc, l'objet à être reconnu. La seconde ligne, donnée par l'algorithme de reconnaissance, équivaut à la vue la plus probable de l'objet reconnue par le K-plus proches voisins. Il est intéressant remarquer que l'invariance à rotation du descripteur trompe l'estimation de l'orientation en prenant son correspond énantiomorphe dans le premier carré rouge. Autrement, le dos du pingouin étant une grande surface presque plane, il est partiellement retiré par l'étape de segmentation. Ainsi, le nuage de points résultant de ce point de vue n'est pas suffisamment complet pour caractériser correctement l'objet, ce qui induit une mauvaise reconnaissance dans le carré bleu. Au final, on remarque que le traitement apporté par la chaîne de Markov cachée permet de corriger les problèmes d'une base de donnée relativement sparse avec des possibles erreurs de segmentation, permettant la correction simultanée de la reconnaissance de l'objet et de son orientation en ligne.

3. L'angle entre la base et la tête de la caméra Asus Xtion est facilement modifié.

Dans le premier tableau on retrouve les résultats de la reconnaissance donné par la comparaison des histogrammes provenant du \*plus proche voisin\*. Ce résultat estime la capacité de distinguer deux objets quelconques, en autre mots, cette capacité viens de la représentativité des descripteurs utilisés et l'efficacité de la mesure de similarité entre histogrammes.

### 3.3.2 Suivi et reconnaissance multi-cibles

Le deuxième expérimentation correspond à placer des objets appris auparavant dans une pièce et conduire le robot faisant en sorte qu'il les regardait de plusieurs points de vues différents. Ce scénario est beaucoup plus complexe que celui d'avant. D'abord les objets occultent uns aux autres résultant en mauvaises segmentations. Ensuite, la suivi d'objets est plus complexe La carte finale donnée par l'algorithme, 3.3,

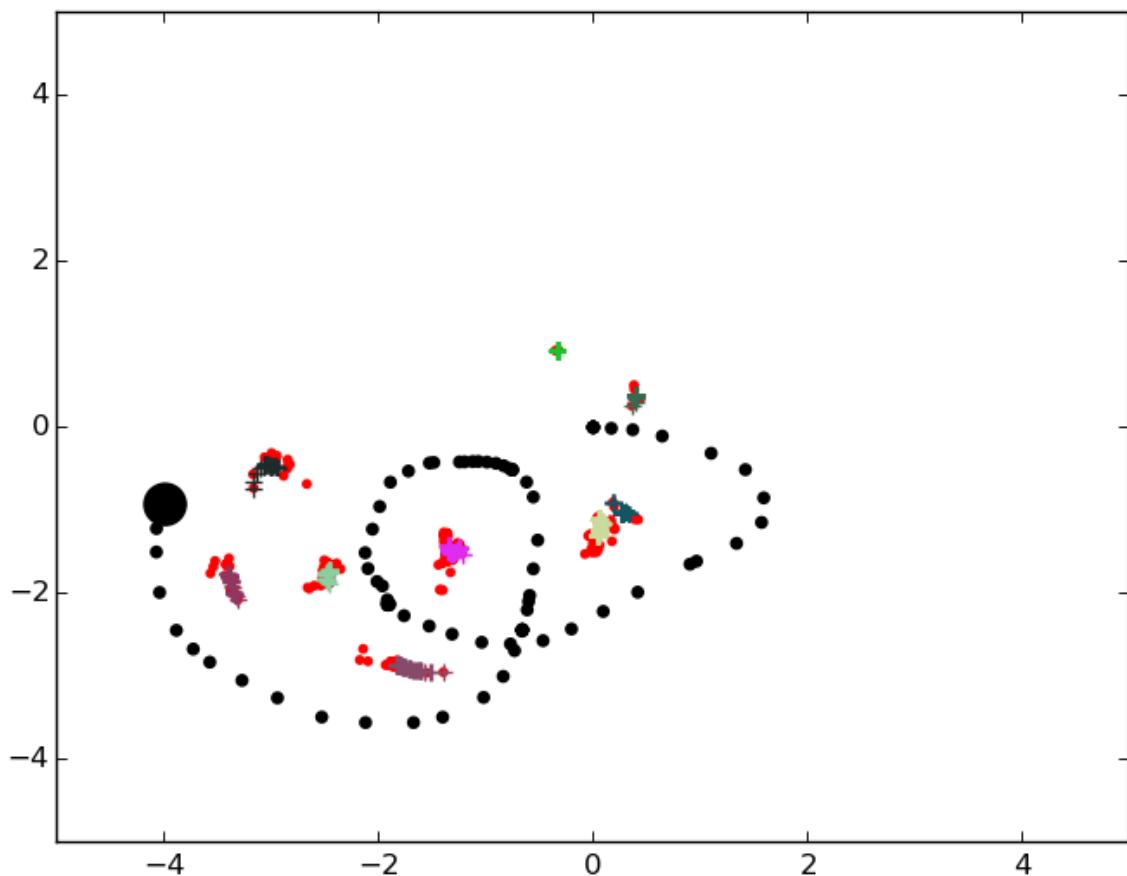


FIGURE 3.3 – ...

## 3.4 Problèmes rencontrés

### Synchronisation

### Problèmes de déplacement

La roulette de support originellement installée avait deux axes de rotation. Pourtant, quelques mouvements de rotation du robot alignent la roulette orthogonalement au sens du prochain mouvement ce qui crées une torche parasite que perturbe la trajectoire voulue. Une tentative frustrée d'installer une bille omnidirectionnelle à roulement, qui se bloquait sur la moquette avec le poids

du robot, a fait que l'originale était réinstallée. Une deuxième solution serait d'interdire certains mouvements du robot pour éviter cette déviation.

# Chapitre 4

## Conclusion

### 4.1 Synthèse

bien La principale contribution de ce projet est liée au traitement de la reconnaissance qui intègre un modèle temporel de transition entre vues. La méthode proposée ici pourrait être mise en place pour n'importe quel système de reconnaissance d'objets à condition qu'il soit mobile et capable de fournir une estimation de son déplacement, et que chaque élément de la base des objets à reconnaître soit associé à une estimation de son orientation. La reconnaissance d'objets multi-vues augmente la capacité à résoudre des situations ambiguës et gère les problèmes de bruit provenant de la base de donnée (absence de vue, erreurs de segmentation). en conséquence ...

Le robot n'a pas de restriction de déplacement pour reconnaître les objets.

### 4.2 Discussion

mauvais  
limitations du système

### 4.3 Perspectives

améliorer Les possibles améliorations seraient d'intégrer un algorithme de SLAM pour rendre plus robustes les estimations de position et avoir une meilleure représentation de l'environnement. On pourrait ajouter ensuite un méthode de planification de trajectoires afin que le robot puisse se déplacer de manière autonome. Par ailleurs, on peut envisager une extension du filtre de Kalman pour des objets en déplacement grâce à des modèles cinématiques suggérés par la reconnaissance.

Quelques améliorations que tu n'as pas citées

- Dans les HMM, mettre une faible probabilité de changer d'objet (en cas d'occlusions, deux objets à côté ...). Associer à cela la possibilité d'ouvrir une nouvelle track si l'objet change
- Renforcer le filtre de Kalman avec une observation de similarité liée aux features, pas seulement à la position x,y de l'objet.
- Ajouter des features de couleur et de texture pour lever les ambiguïtés de vues
- Faire une base incrémentale (à moins que tu ne la fasse la semaine prochaine)

# Bibliographie

- [1] Haider Ali, Faisal Shafait, Eirini Giannakidou, Athena Vakali, Nadia Figueroa, Theodoros Varvadoukas, and Nikolaos Mavridis. Contextual object category recognition for rgb-d scene labeling. *Robotics and Autonomous Systems*, 62(2) :241–256, 2014.
- [2] Ali Borji, Majid Nili Ahmadabadi, and Babak Nadjar Araabi. Learning sequential visual attention control through dynamic state space discretization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2258–2263. IEEE, 2009.
- [3] Hermann Borotschnig, Lucas Paletta, Manfred Prantl, Axel Pinz, et al. Active object recognition in parametric eigenspace. In *BMVC*, pages 1–10. Citeseer, 1998.
- [4] Björn Browatzki, Vadim Tikhanoff, Giorgio Metta, Heinrich H Bülfhoff, and Christian Wallraven. Active object recognition on a humanoid robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2021–2028. IEEE, 2012.
- [5] Gilles Burel and Hugues Hénocq. Three-dimensional invariants and their application to object recognition. *Signal Processing*, 45(1) :1–22, 1995.
- [6] Louis-Charles Caron, David Filliat, and Alexander Gepperth. Neural network fusion of color, depth and location for object instance recognition on a mobile robot. In *Computer Vision-ECCV 2014 Workshops*, pages 791–805. Springer, 2014.
- [7] Guillaume Duceux and David Filliat. Unsupervised and online non-stationary obstacle discovery and modeling using a laser range finder. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 593–599. IEEE, 2014.
- [8] Martin A Fischler and Robert C Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.
- [9] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *Computer Vision-ECCV 2004*, pages 224–237. Springer, 2004.
- [10] Danica Kragic. Object search and localization for an indoor mobile robot. *CIT. Journal of Computing and Information Technology*, 17(1) :67–80, 2009.
- [11] Cédric Le Barz, Nicolas Thome, Matthieu Cord, Stéphane Herbin, and Martial Sanfourche. Global robot ego-localization combining image retrieval and hmm-based filtering. In *6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pages 6–p, 2014.
- [12] Sanjay Dhar Roy, Santanu Chaudhury, and Sean Banerjee. Isolated 3d object recognition through next view planning. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 30(1) :67–76, 2000.

## Annexe A

# Matériels

### A.1 Plateforme mobile

#### Robot Wifibot v2

##### Dimensions :

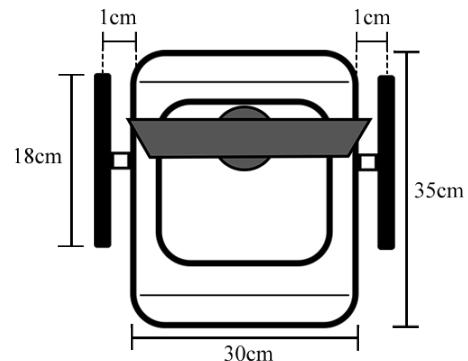
Hauteur : 18 cm

Largeur : 35 cm

Longueur : 30 cm

Distance entre roues : 0.32 cm

Diamètre des roues : 0.18 cm



#### Ordinateur portable embarqué :

HD : 8 Go

RAM : 2 Go

Batterie : 12V NiMH 3.8A 9000mAH

Processeur : Intel® Atom™ N270 @ 1.60GHz

Système opérationnel : Ubuntu 14.04

Version ROS : ROS Indigo



## A.2 Ordinateur Portable

### HP Pavilion g6

Processeur : Intel® Core™ i5-3230M @ 2.60GHz

HD : 750Go

RAM : 4Go

Système opérationnel : Ubuntu 14.04

Version ROS : ROS Indigo



## A.3 Capteur RGB-D

### Asus Xtion PRO LIVE

Distance d'utilisation :

de 0.8 à 3.5 mètres

Range de vision :

58°Horizontal, 45°Vertical, 70°Diagonal

Resolution :

VGA (640x480) : 30 fps

Utilisation intérieur



## Annexe B

# Logiciels

Le design de l'architecture a permis de définir les unités de traitement et la communication entre elles. La définition des unités de traitement suit le découpage du pipeline de reconnaissance avec des nœuds dédiés pour la segmentation, l'extraction de features, la classification, mais également le contrôle du robot.

L'interfaçage matériel-logiciel a été réalisé sur l'environnement ROS - Robot Operating System. En plus d'outils d'affichage, ROS rassemble des bibliothèques d'acquisition d'images RGB-D, OpenNi 2 et Freenect, ainsi qu'une bibliothèque de traitement de nuage de points, PCL.

De plus, sa structure en nœuds a permis une implémentation modulaire et directe du système, ainsi que de gérer la communication entre l'ordinateur portable et le processeur embarqué sur le robot. La structure complète de l'architecture peut être retrouvé dans les annexes ??.

### B.0.1 Restrictions logiciels

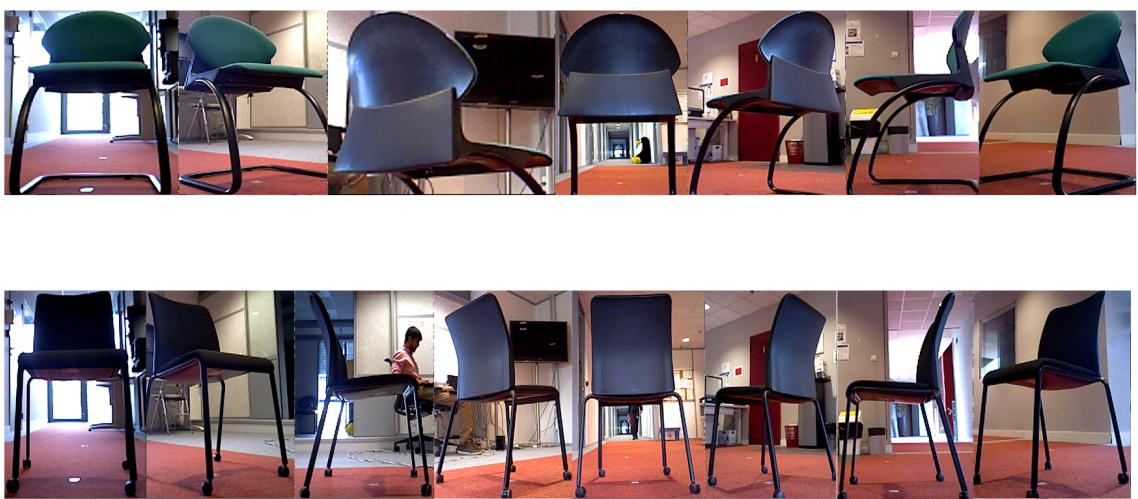
L'ordinateur embarqué a un puissance de calcul réduit ce que ne permet pas que le node d'acquisition *openni2.launch* tourne correctement. La solution pour l'instant c'était de connecter le capteur Asus sur l'ordinateur portable HP.

## Annexe C

### Base de données



FIGURE C.1 – Composition complète de la base de données.



## C.1 Segmentation

### C.1.1 Paramètres

- Distance maximale au capteur : 3 m
  - Distance pour qu'un point soit considéré comme appartenant au plan : 5 cm
  - Taille du *grid* de voxalization : 2 cm
  - Rayon d'estimation de la normale : 2 cm
  - Aire de *smoothing* de la normale :  $10 \text{ cm}^2$
  - Distance minimal du plan du sol pour qu'il soit considéré comme partie de l'objet : 3 cm
- La plus parts de valeurs étaient choisis telle comme ils était proposé dans la librarie PCL.  
Quelque autre ont été modifiés pour atteint caractéristique attendue.

### C.1.2 Floor Detection

The major concern goal of the algorithm is to well estimate the floor plan coordinates. From that, other plans like walls could be inferred, supposing they have a fixed geometrical relation. The RANSAC algorithm provide a reasonable solution to the problem and it is already implemented in the PCL library.

Some parameters need to be set, such as deviation from the plan mathematical model.

The parameters used for the robot are described at the annex section.

### C.1.3 Estimation de la normale

Pour constituer les informations géométriques l'estimation de la normale du point est d'extrême importance.

Sont calcul est fait de la manière suivant : 1. Un nombre de voisins est choisi 2. Ces point \*servent\* à trouver des paramètres de l'équation du plan tangent et, par consequent, la normale correspondent.

Le méthode adopté pour la bibliothéque PCL correspond à prendre un certain nombre de plus proches voisins définis par un seuil. Un petit seuil rende le calcul faux et un grand prend en compte points distants que peuvent ne pas faire partie du plan estimé.

[http://www.pointclouds.org/documentation/tutorials/vfh\\_recognition.php#vfh-recognition](http://www.pointclouds.org/documentation/tutorials/vfh_recognition.php#vfh-recognition)

[http://pointclouds.org/documentation/tutorials/fpfh\\_estimation.php](http://pointclouds.org/documentation/tutorials/fpfh_estimation.php)

<https://github.com/PointCloudLibrary/pcl/wiki/Overview-and-Comparison-of-Features>

### C.1.4 RANSAC

The RANSAC algorithm is a learning method to estimate a given model parameters. Contrary to other estimation algorithms, which considers the whole data representative to model estimation, RANSAC suppose the existance of **inliers** or **consensus** and **outliers** and uses a voting scheme to select between reliable data, that must follow two assumptions :

- Noisy data will not vote consistently for a single model - (few outliers)
- Enough good features voting for the same model - (few missing data)

#### The RANSAC algorithm

The iterative algorithm is composed of two different stages :

- Sample minimal data from dataset requerid to first estimate model parameters.

- Given a threshold error, it selects data points that are consistent to the model created in the first step.
- Random hypothetical inliers subset
- Find model parameters
- All data tested according to a loss function that determine the *consensus*
- Finishes when a sufficient number of point belongs to the *consensus*

If the variables are linear and normally distributed the Bayes filter becomes equal to the Kalman filter.

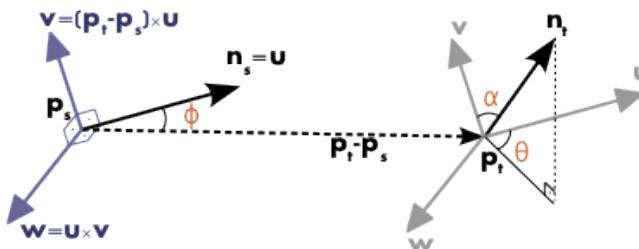
## C.2 Descripteurs

### C.2.1 Point Feature Histogram - PFH

Le PFH incorpore les notions de courbure des objets par le calcul de l'écart entre les normales de points. Ce descripteur peut être calculé localement ou globalement, en changeant l'importance du rayon de comparaison. Il est la base d'une grande famille de descripteurs, desquels quelques-uns seront expliqués dans la suite.

En revenant à son calcul, l'histogramme est évalué à partir des pairs de points à l'intérieur d'un ensemble prédéfini. D'abord, un repère initial, illustré dans l'image \*9\* est établi sachant le vecteur distance normalisé et les deux normales. Ensuite, trois angles, qui correspondent à la transformation angulaire entre les deux normales, et la distance euclidienne entre le deux points sont estimés. Ces quatres valeurs seront considérés comme features pour réduire l'espace initial de douze dimensions - coordonnées et normales des deux points - à un espace de quatre dimension.

$$u = n_s \quad v = u \times \frac{(p_t - p_s)}{\|p_t - p_s\|_2} \quad w = u \times v$$



Puis, les normales sont traduites en features angulaires décrit par les équations :

$$\alpha = v \cdot n_t \quad \phi = u \cdot \frac{(p_t - p_s)}{d} \quad \theta = \arctan(w \cdot n_t, u \cdot n_t) \quad d = \|p_t - p_s\|_2$$

La prochaine étape c'est de calculer l'histogramme en-soi. Un subdivision du range de valeur de chaque feature angulaire, normalisés pour rester dans le même intervalle trigonométrique, est faite et chaque cellule du histogramme est incrémenté dès qu'une feature tombe dans cet intervalle.

Le PFH se présente robuste à des différents échelles de densité de points et de bruit, au même temps que invariant à les transformations affines. Des inconvénients vient de la dépendance de la qualité de l'estimation de la normale<sup>1</sup>.

### C.2.2 Fast Point Feature Histogram - FPFH

L'avènement du FPFH viens de la motivation de réduire la complexité de calcul du descripteur PFH,  $O(nk^2)$ , pour un nuage avec  $n$  points où chaqu'un des points à  $k$  voisins . Pour cela, l'algorithme au lieu de calculer la relation bidirectionnelle entre tous deux points de l'ensemble définis,

1. Une discussion des méthodes présentées sur PCL est mis dans les annexes.

les features de chaque point sont pondérées par les voisins à l'intérieur d'un rayon de recherche, selon la formule au-dessous :

$$FPFH(\mathbf{p}_q) = SPFH(\mathbf{p}_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_k} \cdot SPFH(\mathbf{p}_k)$$

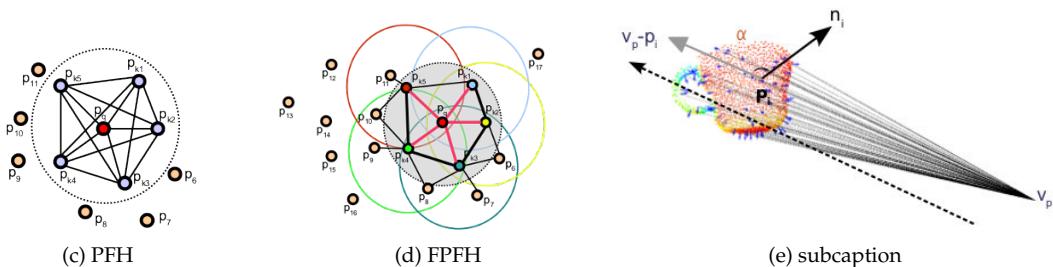
Cette procédure résulte dans une complexité O( $n*k$ ). Le gain en vitesse est considérable, ce qui permet des applications en temps réelles. De plus, pour éviter une perte d'information considérable, le FPFH incorpore quelques points externes au rayon de voisinage, mais qui sont compris dans un rayon de taille.

### C.2.3 Viewpoint Feature Histogram- VFH

Le VFH, différemment du rapport entre PFH et FPFH, c'est une extension du deuxième descripteur où la variance de point de vue est prise en compte. De forme succincte, des angles entre la normale de chaque point et la direction principale d'observation est concaténée à l'histogramme provenant du SPFH (Simplified PFH). En gardant le repère utilisé dans les descripteurs d'avant, le vecteur direction principale est défini par la différence entre l'origine du capteur jusqu'au centre de *cluster*. Ce résultat permet, au même temps, de reconnaître l'objet et son orientation spatiale, et, par conséquent, c'est le feature utilisé dans les premières expériences.

### C.2.4 Clustered Viewpoint Feature Histogram - CVFH

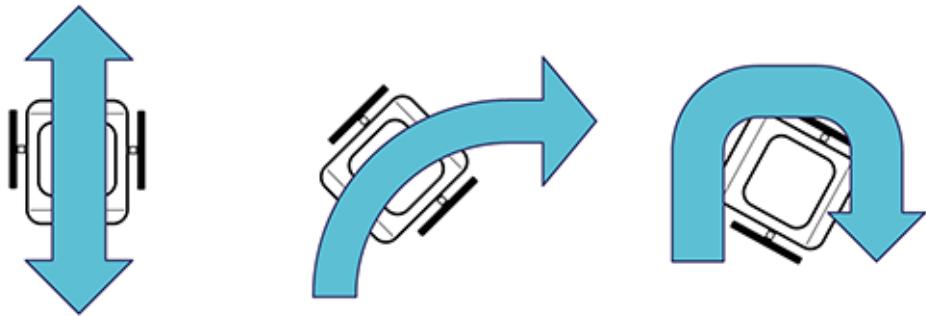
CVFH - Clustered VFH - est une feature semi-global capable de gérer occlusions partiels, mauvaise segmentation et bruit par la décomposition du *cluster*, segmenté comme objet, en sous-clusters de structure spatiale homogène. Le descripteur est obtenu d'après un premier filtrage de zones de haute gradient de courbure, considérés comme zones de transitions entre surfaces, et, puis, et l'estimation de l'histogramme VFH pour chaque surface donnée par l'algorithme *point growing*. Ainsi, pour un seul objet, le CVFH ne génère pas un seul histogramme VFH, mais un vecteur des histogrammes. En revanche, le découpage exige un soin un plus avec la résolution des surfaces pour quelles restent représentatives de l'objet.



L'Universidad de León a fait un compte rendu des *features* implementés sur PCL dans le lien \*[8]\*. Plus d'information sur les descripteurs et leurs implementations sur la librairie PCL peuvent être retrouvés sur le site internet <http://pointclouds.org>.

### C.2.5 Déplacement du robot

Le robot est équipé de trois roues, desquelles les symétriques arrières sont motorisées et responsables pour le déplacement motrice. Au même temps que la dernière sert à donner un support pour la partie derrière du châssis. Les moteurs sont contrôlés à partir de commandes série, préétablies pour le fabricant, qui définissent la vitesse de roulement. La combinaison des rotations des deux roues motorisées dans les deux sens possibles permet au robot d'avoir les comportements suivants :



- Déplacement en ligne droite : deux roues roulant avec la même vitesse et dans le même sens.
- Déplacement en arc de cercle : différence entre les vitesses des roues.
- Rotation : deux roues à la même vitesse, mais avec de sens différents.

Finalement, la combinaison de ces mouvements permet au robot de accéder n'importe quelle position de l'espace.

### C.2.6 Fusion de données

L'estimation de l'odométrie diverge au long du temps dû à l'accumulation d'erreurs mesure. Cette divergence est encore plus considérable . Dans l'autre côté, l'utilisation du senseur RGB-D estime la distance au centroïde de chaque objet. Une correspondance entre les objets de deux observations consécutives nous donné une autre repère de positionnement. Un couplage des deux mesure, une provenant de l'encodeur moteur et l'autre du capteur infrarouge, fait que l'odométrie doive être

## C.3 Planning de Travail

- Mise en place de l'architecture et des protocoles de communication entre composants physiques.
- Étude bibliographique initiale pour situer le travail par rapport à l'existant.
- Implémentation de l'asservissement d'une caméra PTZ par rapport au retour d'un algorithme de tracking.
- Aperçu de certaines limitations de la caméra PTZ qui a été remplacée par une caméra RGB-D.
- Utilisation d'un algorithme de segmentation d'objets possibles dans la scène.
- Asservissement en boucle ouverte du robot pour la création de la base de données.
- Premiers tests pour l'acquisition de la base des données.
- Résolution des problèmes trouvés lors des premiers tests pour la création de la base de données.
- Validation de la base de données. Représentativité et reproduction.
- Étude approfondie de l'état de l'art des modèles et méthodes qui puissent être utiles pour notre problème.
- Mise en place de la solution et du modèle proposé.
- Premiers tests et ajustements nécessaires.
- Mise en œuvre de la solution complète.
- Validation finale.

## **Annexe D**

## **Annexe II**

### **D.1 Le groupe Thales**

Les origines du groupe remontent à 1968 avec la fusion de la Compagnie Générale de Télégraphie sans Fil et des activités d'électronique professionnelle de Thomson-Brandt. Cette fusion donne naissance à Thomson-CSF. Dès 1987, l'entreprise entame une restructuration en profondeur de ses activités et met en place une stratégie d'expansion vers l'Europe. En 1998, le gouvernement français cède une partie de ses actions aux sociétés Aerospatiale, Alcatel et Dassault. Le groupe bascule alors dans le secteur privé, cela entraîne aussi une expansion des activités, notamment dans le secteur de la défense, au-delà de l'Europe, comme en Australie, en Corée ou à Singapour. Les activités se sont aussi diversifiées et s'articulent principalement autour de la défense, l'aéronautique et les technologies de l'information. En 2000, Thomson-CSF devient Thales. Le groupe devient un leader dans les domaines de la défense et de l'aéronautique et renforce sa présence dans le domaine de la sécurité civile. En 2009, Dassault devient l'actionnaire majoritaire du groupe en rachetant les parts d'Alcatel. De ce fait, en 2010 l'organisation de Thales est modifiée suivant un système basé sur 3 zones géographiques et 7 divisions afin de simplifier son fonctionnement et améliorer ses performances.

### **D.2 Secteurs d'activité**

Thales est un groupe d'électronique spécialisé dans l'aérospatial, la défense et les technologies de l'information. Coté à la bourse de Paris, présent dans 56 pays et employant 66 500 collaborateurs, Thales est un des leaders mondiaux des systèmes d'information critiques sur les marchés de l'aéronautique et de l'espace, de la défense et de la sécurité. Avec environ 14,2 milliards d'euros de chiffre d'affaire en 2013, le capital du groupe est détenu à 27% par l'État français, 26% par Dassault Aviation et les 47% restants sont flottants. Le portefeuille du groupe est équilibré avec 55% de commandes dédiées à la Défense et 40% au Civil. L'innovation constitue un secteur important pour Thales. Aujourd'hui elle dépasse le seul cadre technologique pour irriguer tous les champs de l'entreprise, de la recherche et développement à l'activité commerciale. Les dépenses de recherche et développement représentent 20% de l'activité du groupe. Avec plus de 25 000 chercheurs et ingénieurs, un portefeuille regroupant 15000 brevets et plus de 30 accords de coopération avec des universités et des laboratoires publics en Europe, aux États-Unis et en Asie, Thales occupe une place de référence dans les domaines de la haute technologie et de l'innovation.

Les travaux de recherche amont sont essentiellement conduits au sein de Thales Research & Technology (TRT), centre de recherche du groupe Thales en France, qui regroupe environ 500 experts autour de trois domaines techniques clés :

- Électronique, électromagnétisme et optronique
- Logiciel et système d'information
- Sciences de l'information et de la cognition

Et dont les activités s'opèrent au sein de sept laboratoires :

- Ingénierie des systèmes logiciels
- Analyse des sources d'information
- Sécurité sur Internet
- Recherche en infra-rouge et imagerie polarimétrique
- Dualité et technologies de souveraineté
- Sécurité biologique et chimique
- Nano-magnétisme

Les liens tissés entre ces équipes de recherche et les communautés académique, scientifique et industrielle, se mettent en place grâce à l'implantation des laboratoires de TRT dans des campus universitaires. En France, c'est le cas du site de Palaiseau qui est implanté sur le campus de l'Ecole Polytechnique.

## D.3 Présentation de ThereSIS

Au sein de la branche « Systèmes d'information et de Communications Sécurisés », se trouve la filiale Thales Service SAS qui travaille sur la conception, le développement et l'intégration des systèmes d'information critiques pour les entreprises et les gouvernements.

C'est à l'intérieur même de Thales Service SAS que se trouve le laboratoire d'innovation ThereSIS (Thales European Research centre for Security & Information Systems). Ce laboratoire est né en Septembre 2006 d'une volonté de renforcer le leadership de Thales dans le domaine particulier de l'ICT vis-à-vis notamment de la communauté européenne. Ce laboratoire de recherches appliquées est l'un des quatre laboratoires de GBU SIX, dédiés aux Etudes Amont, avec TAI (Technologie Avancées de l'Information), SC2 (Software Core) et le CENTAI (Centre d'Excellence Nouvelles Techniques Analyse de l'Information). Un des objectifs communs est de développer des différentiateurs techniques au bénéfice des unités opérationnelles de la GBU et plus largement du groupe Thales. L'équipe initiale était composée de 20 experts en système de sécurité d'information critique. Ensuite, le concept fut étendu au domaine de la "Sécurité Physique" en 2007, et le laboratoire a vu son effectif s'élever à 45 employés. Aujourd'hui l'équipe de ThereSIS compte environ 70 personnes.

## D.4 Secteurs d'activité

Installé à Palaiseau, dans les locaux de TRT, ThereSIS développe des solutions innovantes dans le domaine de la sécurité et de la protection des infrastructures critiques, telles que les aéroports, les centrales nucléaires, les gares, etc. Ces activités se concentrent aujourd'hui sur les sujets suivants :

- La sécurité physique avec le développement de systèmes à base de capteurs innovants, le traitement intelligent de la vidéo, la gestion de crise et l'interopérabilité des systèmes.
- Les mécanismes et les services de sécurité de système d'information et le management d'identités.
- La supervision de la cyber-sécurité des architectures critiques et l'aide à la décision adaptable aux contextes métiers.
- Les modèles, outils et services de sécurité et de management de la multi-conformité en dynamique pour les architectures de type SOA.
- La sécurisation et la supervision des architectures de service vitalisées et le "cloud computing"
- Les interfaces multimodales et les dialogues hommes-machines.
- La modélisation directement exécutable de processus complexes, leur interface graphique et leur sémantique.
- Les environnements synthétiques et leurs apports pour les systèmes d'information critique avec la simulation des comportements humains.

## D.5 Le laboratoire Video Technologies & New Sensors

Le laboratoire Video Technologies & New Sensors est composé de deux domaines : l'analyse vidéo et les systèmes de perception. L'analyse vidéo traite en particulier du développement d'algorithme avancés de traitement d'image via un laboratoire commun entre Thales et le CEA<sup>1</sup>, baptisé VisionLab.

---

1. Commissariat à l'Énergie Atomique et aux Énergies Alternatives.