

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

PROJET DE FIN D'ÉTUDES

---

**Stratégie d'asservissement de robot et de  
caméra**

---

**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 Service | Campus Thérésia  
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>5</b>
1.1	Thématiques du stage. . . . .	5
1.2	État de l'art . . . . .	5
1.2.1	characteristiques qu'on veut avoir, inspiration humaine . . . . .	6
<b>2</b>	<b>Description du Travail</b>	<b>7</b>
2.1	Segmentation . . . . .	7
2.1.1	Algorithme . . . . .	8
2.1.2	Restrictions . . . . .	8
2.2	Descripteurs . . . . .	9
2.2.1	Point Feature Histogram - PFH . . . . .	9
2.2.2	Fast Point Feature Histogram - FPFH . . . . .	10
2.2.3	Viewpoint Feature Histogram- VFH . . . . .	10
2.2.4	Clustered Viewpoint Feature Histogram - CVFH . . . . .	10
2.3	Classification . . . . .	10
2.4	Représentation de l'objet . . . . .	11
2.4.1	Principes de la Reconnaissance Humaine . . . . .	11
2.4.2	Caractéristique des objets . . . . .	11
2.5	Graphe d'aspect polaire . . . . .	12
2.6	Filtre de Kalman . . . . .	13
2.7	Chaînes de Markov Cachées . . . . .	13
2.8	Déplacement du robot . . . . .	14
2.8.1	Estimation de l'odométrie . . . . .	14
2.8.2	Problèmes de déplacement . . . . .	14
2.8.3	Fusion de données . . . . .	14
2.9	Architecture . . . . .	15
2.10	Expériment . . . . .	16
<b>3</b>	<b>Conclusion</b>	<b>17</b>
3.1	Améliorations . . . . .	17
<b>4</b>	<b>Bibliographie</b>	<b>18</b>
4.1	Matériels . . . . .	20
4.1.1	Types d'outils et méthodes utilisés. . . . .	20
4.1.2	Outils matériels et logiciels . . . . .	20
4.1.3	Description de la plateforme mobile . . . . .	20
4.1.4	Description Ordinateur Portable . . . . .	20
4.1.5	Restrictions logiciels . . . . .	20
4.2	Floor Detection . . . . .	20
4.2.1	Estimation de la normale . . . . .	21
4.2.2	RANSAC . . . . .	22
4.3	Planning de Travail . . . . .	22

# Chapitre 1

## Introduction

EN VERT SONT DES PARTIES QUE NE SONT PAS ENCORE FINI

La perception de l'environnement par machines est indispensable pour son intégration à la vie quotidienne. Compétences telle comme se localiser, prise de décisions et capacité d'apprentissage sont nécessaire pour la réalisation de les plus simples tâches. Dans cet étude, on s'intéresse à la compréhension d'éléments constituent d'une scène, sujet récurrent dans le domaine de la vision par ordinateur. Plusieurs approches proposés dans la littérature explorent une sous-partie du *pipeline* de la reconnaissance pour faire face au difficile défi de représenter la forme visuelle des objets, pendant qu'autres s'intéressent à l'utilisation du système de reconnaissance pour tâches de recherche dans l'environnement et saisie (*empoignement*), par exemple. Notre démarche correspond, initialement, à l'intégration de techniques de l'état de l'art pour arriver à un système fonctionnel de reconnaissance incorporé sur une plateforme mobile équipée d'un capteur RGB-D capable. Dans une première temps, la plateforme est capable d'acquérir une base de données d'images d'un objet et, ensuite, d'utiliser ces informations apprises pour vérifier si un objet candidat est ou non celui présenté auparavant, et, finalement, renforcer sa perception avec les information de son déplacement pour lever les possibles ambiguïtés.

### 1.1 Thématiques du stage.

Vision par ordinateur - Reconnaissance d'objets multi-vue - Robotique

### 1.2 État de l'art

La problématique de la recherche d'objets par un robot mobile est traitée dans l'article *Object Search and Localization for an Indoor Mobile Robot* [1], dont, l'intérêt central, est d'effectuer une exploration dans un environnement où le plan est généré par avance avec des capteur laser et du SLAM, ce qui permet la discréétisation en zones de recherche, pour retrouver un objet donné a priori. Le système est capable de retrouver des objets dans des environnements assez complexes, en s'utilisant des descripteurs SIFTs pour la reconnaissance et des heuristiques pour l'estimation de la distance.

Dans l'autre côté, le travail *Active Object Recognition in Parametric Eigenspace* s'intéresse à la reconnaissance d'objet active avec traitement de vues ambiguës par le repositionnement de la camera. Une étape initial d'entraînement, transforme les images d'objets dans la base du espace propre et la reconnaissance est fait par le projection d'une image de teste dans cet espace. Cela apporte aussi une estimation de l'orientation de l'objet, une fois que les images sont labellisées. Pour décider la prochaine action, l'algorithme propose la réduction de l'entropie en s'utilisant des probabilités de reconnaissance à partir des informations antérieures.

Autrement, l'algorithme *Next Best View* traite l'optimisation du nombre d'actions nécessaires par un bras mécanique de six degrés de libertés pour réparer des surfaces chachées et des événements visuels d'un objet à fin d'atteindre un modèle CAD à 3 dimensions. Pour un robot terrestre de déplacement contrainte en deux dimensions tout l'information concernant un objet peut être aquis en faisant un tour complèt autour du même, ce qui réduit la complexité du espace de recherche. La distance, aussi, joue un rôle important lorsque la résolution des capteurs est limité, mais la capacité de se rapprocher compense ce handicap.

Dans, une autre perspective, la reconnaissance d'objet est une tâche évident pour des humans et partir dessous sers comme font d'inspiration. Selon les études réalisés en **référence [5]**, les humans semblent construire un modèle géométrique rotationnel continu, en autre mots, une composition de vues séquentielles qui font le lien entre différentes faces du objet.

L'œuvre *Neural Network Fusion of Color, Depth and Location for Object Instance Recognition on a Mobile Robot* part d'un algorithme de segmentation, à partir de l'extraction de plans correspondant aux murs et au sol, pour ensuite classifier la concaténation de descripteurs - PFH, SIFT et RGB - de l'objet par un *multi-layer perceptron* où la sortie correspond à une couche *softmax*. Dans un premier temps, le classificateur est entraîné de façon supervisée vers l'algorithme *backpropagation* dans une base d'objets initiale, pour, ensuite, à l'aide de la cartographie et localisation simultanée, étiqueter les objets présents dans l'environnement.

*Active Object Recognition on a Humanoid Robot* traité la reconnaissance d'objets multi-vues pour un robot humanoïde comme un problème de localisation et suivi par un filtre particulier.

Au terme de l'analyse, l'ouvrage *Three-Dimensional Object Recognition and 6 DoF Pose Estimation* décrit une méthode simultanée d'estimation de pose et reconnaissance d'objet selon deux types perspectives de traitements : global et local, avec un compte rendu des leur respectives descripteurs. L'article suggère une segmentation, nécessaire dans le traitement global, simple qui considère les objets comme des points sur un plan support. L'étape de correspondance entre vues est faite par le classificateur K-plus proche voisin

### 1.2.1 caractéristiques qu'on veut avoir, inspiration humaine

C'est continuité espaciel est, donc, une caractéristique fortement souhaitable. Cette continuité est représentée dans l'article **TODO** où un graphe d'aspect hiérarchique est construit d'après la variation de features. Autre étude \*réf\* suggère que l'ensemble de vues augmente le taux de reconnaissance des objets chez les humans, ce qui paraît intuitif.

#### 1. View-based dynamic object recognition based on human perception

Cet article élaboré l'importance de une observation temporelle pour la reconnaissance des objets, fait remarqué dans le processus de reconnaissance humaine. La méthode utilisé consiste à partir de l'image initial prise comme key-frame où ses features sont calculées. Ensuite, ses features sont trackées jusqu'à diminution de features détectées tombe à un nombre inférieur à un seuil défini à priori. La reconnaissance est faite par le comparaison de

## Chapitre 2

# Description du Travail

La majorité de la littérature traite le problème de la reconnaissance d'après une seule image de l'objet. Typiquement, une ensemble de *features* est extrait et, ensuite, comparé aux modèles d'objets présents dans une base de données initiale, en contraste aux méthodes directes, comme deep learning, où l'image d'entrée est associée directement avec des classes des objets correspondants au compromis d'une étape d'entraînement importante, pour l'apprentissage de *features*, encore plus dans un espace à 3 dimensions provenant du capteur RGB-D.

Un grand effort était fait pour améliorer l'extraction, le *matching*, ainsi que le *features* elles-mêmes pour qu'elles soient invariantes à transformations affinés de l'image et représentatives de l'objet. Ce traitement classique a l'avantage d'être, à la fois modulaire, avec des étapes bien définies de segmentation, Extraction de features, classification et post-traitement, et, au même temps, d'avoir des résultats satisfaisants d'après une implémentation plus immédiate. Malgré son intérêt dans certains cas, rapidement on s'aperçoit de limitations lors que vues ambiguës apparaissent.

L'utilisation d'un algorithme de reconnaissance basée sur une seule image apporte l'inconvénient de n'incorporer pas les notions de vue et de transition entre elles, au contraire, la majorité de ces systèmes souhaitent être invariant à les vues d'objets, en autres mots, avoir la capacité de l'identifier de n'importe quel point de vue. Un système dérivé de celui-ci pourrait traiter le concept de vues plus représentatives et transitions, par contre, de façon moins intuitive. De cette manière, les articles présentés auparavant travaillent sur le domaine multi vue, incorporant des aspects géométriques, pour augmenter la qualité de son estimation.

En dernière analyse, l'objectif ultime c'est d'avoir une reconnaissance multi vue, en instance, capable d'incorporer son déplacement pour résoudre des ambiguïtés et faux positifs. Pour incorporer les notions voulus, on présente, simultanément, un simple modèle d'objet suffisamment général et un système capable d'estimer l'orientation de l'objet reconnu, ou bien un système de reconnaissance de vue, pour, ensuite traiter l'information motrice du robot pour augmenter le taux de réussite.

### 2.1 Segmentation

La segmentation consiste de la soustraction des objets d'une image brute, en autres mots, différencier les éléments non constituent du objet de lui-même. La compréhension de la continuité des objets est considérée comme un défi majeur dans le traitement d'image étant donné qu'une fois l'objet séparé du fond, la reconnaissance devient beaucoup plus évident. Une énorme partie de sa difficulté vient du fait de la projection de la scène dans le plan supprimer l'information correspondant à distance. Les capteurs stéréoscopiques et infra-rouges ont recomposé cette absence d'information et simplifié énormément le traitement nécessaire pour obtenir des objets potentiels. Les cartes de profondeur pourraient être utilisées pour représenter cette nouvelle information, pourtant, encore plus naturelle, le concept de nuage de points propose une représentation spatial en trois dimensions de

l'environnement capturé.

La démarche proposée par la littérature considère les objets comme des ensembles de points définis par un seuil initial de proximité. Cette définition est bien extensive et permet de représenter une énormité, sinon tous, les objets. Néanmoins, définir ces ensembles dans une image brute n'est pas tout à fait simple. En conséquence, un nouveau à priori qui spécifie que les objets se placent sur des plans de support, malgré plus restrictif que la définition d'avant, permet un segmentation crédible.

### 2.1.1 Algorithmes

La méthode de segmentation de l'algorithme Tabletop se base exactement sur ces aprioris. Pour retrouver les objets posés sur une table, l'algorithme recherche récursivement les plans de support, où le plus important est pris comme la table. Autrement, l'article ENSTA, en partant du même principe, propose un traitement pour le fond de la scène, où les plans orthogonaux à normale du sol et de taille suffisamment grande sont considérés comme des murs, orientés à segmentation d'objets dans les environnements intérieurs. Ainsi, la dernière segmentation, proposé par \*Luis Charles\*, répond aux exigences du domaine de déplacement du robot : le laboratoire de Thales, Therésis.

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

1. Soustraction du sol... **DETAILLER**
2. Filtrage de points distants, considérés comme plus incertains.
3. Calcul de la normale des superficies comprises dans la scène
4. Élimination de murs, considérés comme de plans orthogonaux au sol de taille suffisamment grande, d'après un seuil.
5. Projection des points appartenant aux objets dans le plan du sol.
6. Détermination de l'enveloppe convexe correspondant au sol détecté.
7. Réduction de la densité de discréétisation pour accélérer l'étape de *clustering*.
8. Clustering des objets par l'algorithme \*point growing\*
9. Retour à discréétisation initiale.
10. Calcul du centroïde et *bouding boxes* 2D et 3D
11. extraction d'imagettes, et autre informations pertinentes aux objets détectés.

Une calibration initiale est nécessaire pour définir l'équation du sol. Pour cela, on place le robot dans un endroit de façon que l'image aperçue correspond majoritairement au sol. L'équation du plan plus important, plus grand nombre de points dans le nuage, est extrait par le RANSAC et sauvegardé dans un fichier texte. Une explication plus détaillée sur les sous-méthodes utilisées pour chaque étape, telle comme le RANSAC est présentée dans les annexes, ainsi comme une discussion des paramètres utilisés.

### 2.1.2 Restrictions

Les physiques de capteurs restreins les types d'objet qui peuvent être aperçus et, ensuite, segmentés, soit à cause de l'interaction avec les rayons infra-rouges, soit à cause de résolution limitée des images mesurées. Dans l'autre côté, la segmentation a ses propres contraints concernant le positionnement des objets dans l'image et, principalement, la définition de sol et murs, résultant dans les restrictions suivantes :

- L'objet se trouve par terre.
- L'objet se trouve au centre de l'image
- Ambiant isolés de lumière infra-rouge
- 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.

– L'objet n'est ni transparents et ni trop réflective.

Un grand nombre d'objets, entre chaises tables, écrans, boîtes en carton, poubelles, de tailles et formes variés étaient testés et peuvent être segmentés malgré les restrictions listées. Quelques exemples de segmentation sont présentés dans les annexes pour illustrer la capacité de segmentation.

## 2.2 Descripteurs

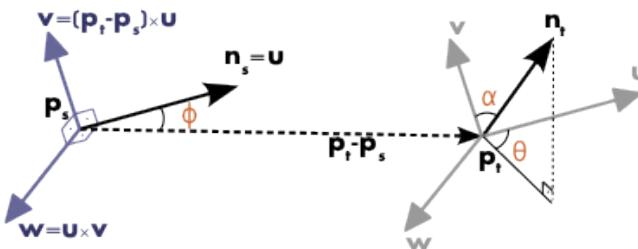
Le travail des descripteurs est, d'un côté, de comprendre les caractéristiques intéressantes et, d'un autre, de réduire la dimensionnalité du espace traité, tandis que restant robuste à des transformations affines et changement de luminosité. On classifie les descripteurs selon la caractéristique qu'il exprime. Une première group sont les descripteurs géométriques qui essaient de traduire les idées de courbure, forme et taille dans histogrammes, et sont intéressant pour étudier les ambiguïtés de reconnaissance, un fois que la plus parts d'objet ont une certaine symétrie spatiel.

### 2.2.1 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 établis 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 point - à un espace de quatre dimension.

$$u = \mathbf{n}_s \quad v = u \times \frac{(\mathbf{p}_t - \mathbf{p}_s)}{\|\mathbf{p}_t - \mathbf{p}_s\|_2} \quad w = u \times v$$



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

$$\alpha = v \cdot \mathbf{n}_t \quad \phi = u \cdot \frac{(\mathbf{p}_t - \mathbf{p}_s)}{d} \quad \theta = \arctan(w \cdot \mathbf{n}_t, u \cdot \mathbf{n}_t) \quad d = \|\mathbf{p}_t - \mathbf{p}_s\|_2$$

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>.

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

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

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

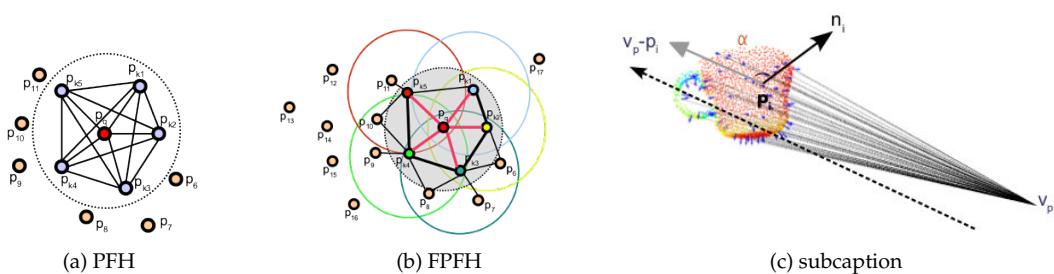
Cette procédure résulte dans une complexité  $O(n*k)$ . Le gain en vitesse est considérable, ce qui permet des applications en temps réelles. De plus, pour éviter une perte d'information considérable, le FPFH incorpore quelques point externes au rayon de voisinage, mais que sont compris dans un rayon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

## 2.2.3 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 centroïde du *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.

## 2.2.4 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 généré 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.



## 2.3 Classification

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 similarité 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.

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.

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

## 2.4 Représentation de l'objet

Ces choix débouchent sur un système fonctionnel de reconnaissance de vue qui permet de s'intéresser, ensuite, par le couplage de résultat de la reconnaissance avec les informations de déplacement du robot.

### 2.4.1 Principes de la Reconnaissance Humaine

Commençant par le modèle de l'objet, le but c'est de intégrer et respecter certains principes appris après observation dans la reconnaissance chez les humains :

1. Gazltat : Tendance à retrouver des formes et contours simples et naturels par regroupement de caractéristiques et/ou comportements.
2. Continuité : l'apprentissage d'un nouvel objet se fait de forme continue. Dans le cas discret, cela revient à un modèle qui simule les transitions entre superficies.
3. Temporalité et séquentialité : Des études ref suggèrent que l'ordre de visualisation de surfaces des objets influence sa reconnaissance à posteriori. Par conséquent, la séquence spatiale entre vues joue un rôle sur le concept d'objet, où parcourir séquence dans la même ordre que celle appris apporterais plus d'information.

Malheureusement, avoir tous ces principes est une tâche assez complexe pour l'état courant de la technologie, pourtant, en même temps, ils inspirent possibles solutions et représentations. L'apport de cet étude se place dans les domaines de la temporalité et séquentialité.

### 2.4.2 Caractéristique des objets

En regardant dans la perspective des objets, certaines de ses caractéristiques sont utiles pour le différencier un des autres :

1. Taille
2. Position global
3. couleur et texture
4. Contraintes d'espace
5. Contexte dans l'environnement
6. Forme géométrique :
  - Sous formes primaire
  - Position et orientation relatif entre formes primaires
7. Affordance : se réfère au concept d'interactions possibles avec un objet. De manière illustratif, dans le cadre du robot utilisé, cela reviendrait à capacité de pousser un certain objet, d'où l'intérêt de l'identification de l'orientation de l'objet.

Le modèle proposé doit être capable d'exprimer au mieux ses caractéristiques en restant, encore, simple. En reprenant la discussion de l'état de l'art, on présent quelques modèles usuellement utilisés pour représenter les objets en trois dimensions.

### Modèle CAD

Consiste à représenter l'objet par son modèle 3D fait à l'aide d'outils de design numériques. L'avantage vient du fait d'une fois le modèle construit, la visualisation de l'objet de n'importe quel vue devient évident. De l'autre côté, la fiabilité du modèle est intérieurement lié à la précision de la reconstruction 3D de l'objet, où un soin avec l'échelle et dimensions, ainsi que avec la reproduction de la couleur et texture, est important pour la bonne représentativité.

### Évolution de contours

Une autre approche est basé sur les silluettes des objets et leur évolution d'après transformation affines. Cette problématique c'est démontre mathématiquement compliqué au niveau de la modélisation de fonctions de contour et de leur transformation. Cependant, une fois modélisé, une prévision

### Squelettes

...

### Aspect-Graphs

Cette forme de représenter les objets consiste à avoir un graphe où chaque noeud correspond à une image d'un point de vue et les liens entre noeuds les réelles transitions visuelles. Comme avoir un graphe complet, qui s'approche du continu, apporte une besoin mémoire important et une certaine redondance d'information, la préoccupation principale est de trouver des points de vues représentatives, nommés *key-frames*, qui peuvent être choisis avec politiques suivants :

1. Aléatoire : Ces key-frames peuvent être choisies de forme complètement aléatoire. Absence de calcul intermédiaire ou prétraitement.
2. Intervalle constant : Une façon simple c'est de conditionner les *key-frames* à un écart angulaire fixe. Cela permet d'unifier le nombre de frames pour chaque objet, ce qui peut être intéressant pour certaines applications
3. Événement visuels : Cela correspond à déterminer des grands variations d'intensité des features pour estimer les key-frames plus représentatifs de l'objet. L'inconvénient vient du besoin d'un prétraitement, en plus, orienté différemment pour chaque feature, lors de la création de la base de données.

## 2.5 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.

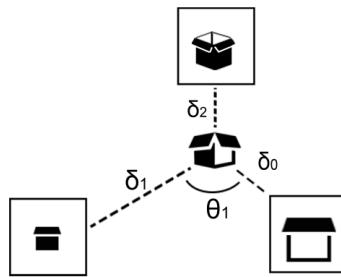


FIGURE 2.1 – Fonction à être optimisé et la solution voulu.

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 \*7\*. 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
- 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.

## 2.6 Filtre de Kalman

La modélisation des objets entraîne le besoin initiale 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 centroïde 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.

## 2.7 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 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, 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.

## 2.8 Déplacement du robot

Le robot est équipé de trois roues, desquelles les symétriques arrières sont motorisés 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, préétablis 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 : équivalent aux deux roues roulant avec la même vitesse et dans le même sens.
- Déplacement en arc de cercle : La différence entre les vitesses de roues résulte dans un mouvement de cercle. Le rapport entre cette différence permettre définir la courbature de la trajectoire.
- Rotation : Dans ce cas, les deux roues sont commandées à la même vitesse, mais avec de sens différents.

\*Illustration\*

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

### 2.8.1 Estimation de l'odométrie

Certains robots sont dotés de capteurs aptes d'estimer de façon approximé sont 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 global du robot.

### 2.8.2 Problèmes de déplacement

La roulette de support originale 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.

### 2.8.3 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

## 2.9 Architecture

Le design de l'architecture sert à décider comment 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 responsables pour la segmentation, calcul de features et la classification, aussi comme, le contrôle du robot.

L'interfaçage matériaux-logiciel était fait sur l'environnement ROS - Robot Operating System. Aussi que les outils d'affichage, ROS, rassemble les bibliothèques d'acquisition d'images RGB-D, OpenNi 2 et Freenect, et de traitement de nuage de point, PCL. De même, sa structure de nœuds a permis une implémentation modulaire et direct du système décrit au-dessus, bien comme la communication entre machines, l'ordinateur portable et celui embarqué au robot.

L'image \*10\* illustre l'architecture en soulignant le flux d'information à travers des nœuds.

## 2.10 Expériment

D'abord, vingt objets de tailles et formes diverses ont été choisi pour évaluer les capacités \*récognitif\* du robot. Ils sont objets typiques qui peuvent être facilement retrouvés dans un laboratoire. Une liste avec tous est incorporé aux annexes. Ensuite, le feature VFH était calculé pour huit positions différents écartées de 45 dégrés. La position correspondant au angle zero, était choisi de manière aléatoire en alignant un des axes de l'objet avec celui du capteur.



Une première évaluation proposée consiste à faire un tour complet autour de l'objet à être reconnu en quatre positions angulaires différentes : 0, 45, 90 et une dernière choisi 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 consiste d'environ 25 images d'angle différent et prendre  $25 \pm 3$  seconds. Ensuite, trois différentes ratios sont calculés pour exprimer la reconnaissance d'objets, la reconnaissance de vue et la suivi des reconnaissances par la chaîne de Markov cachée.

Dans le premier tableau on retrouve le 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.

# **Chapitre 3**

## **Conclusion**

L'apport de ce projet \*relie\* se place dans la partie finale du traitement/pipeline de la reconnaissance. Au remarque que le pos-traitement proposé pourrait être fait pour n'importe quel système de reconnaissance avec estimation de l'orientation du objet reconnu et un système mobile capable d'informer son déplacement. La reconnaissance d'objet multi-vue augment la capacité de résoudre situations d'ambiguité et gère les problèmes provenants de l'absence de vues base de données et erreurs de segmentation, en conséquence ...

### **3.1 Améliorations**

...

## Chapitre 4

# Bibliographie

- [1] Active Object Recognition in Parametric Eigenspace
- [2] View-based dynamic object recognition based on human perception
- [3] L.-C. Caron, D. Filliat, A. Gepperth. Neural Network Fusion of Color, Depth and Location for Object Instance Recognition on a Mobile Robot. Second Workshop on Assistive Computer Vision and Robotics (ACVR), in conjunction with European Conference on Computer Vision, Sep 2014, Zurich, Switzerland.
- [4] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, John Hsu, Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram
- [5] G. Burel and H. Henocq, Three-dimensional invariants and their application to object recognition," Signal Process., vol. 45, no. 1, pp. 1–22, 1995.
- [6] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1999.
- [7] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature Maps for Local Shape Comparison," in SMI '05 : Proceedings of the International Conference on Shape Modeling and Applications 2005 (SMI' 05), 2005, pp. 246–255.
- [8] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in ICRA, 2009.
- [9] B.-C. M. and G. C., "Characterizing shape using conformal factors," in Eurographics Workshop on 3D Object Retrieval, 2008.
- [10] K. Lai, L., X. Ren, & D. Fox A Large-Scale Hierarchical Multi-View RGB-D Object Dataset In IEEE International Conference on Robotics and Automation (ICRA), May 2011.
- [11] R. B. Rusu and S. Cousins, "3D is here : Point Cloud Library (PCL)," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), Shanghai, China, May 9–13, 2011, pp. 1–4.
- [12] Three Dimensional object recognition and 6 dof pose estimation

Sites internet

[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>

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

[http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI\\_tutorial\\_4:\\_3D\\_object\\_recognition\\_\(descriptors\)](http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI_tutorial_4:_3D_object_recognition_(descriptors))

## 4.1 Matériels

### 4.1.1 Types d'outils et méthodes utilisés.

### 4.1.2 Outils matériels et logiciels

En ce qui concerne les aspects matériels, le robot bimoteur Wifibot v2, qui transporte un ordinateur embarqué, sera utilisé comme plateforme mobile. L'acquisition des données est faite par une caméra RGB-D portée par une tourelle qui permet son orientation indépendamment du positionnement du robot. Par rapport au choix logiciel, l'environnement robotique ROS a été adopté pour avoir les deux bibliothèques pour gérer les nuages de points, bibliothèques Freenect et PCL - Point Cloud Library, et d'autres nombreux outils de contrôle du robot et sauvegarde d'informations.

### 4.1.3 Description de la plateforme mobile

#### Robot Wifibot v2

Largeur : 35 cm

Longueur : 30 cm

Hauteur :

Ordinateur portable embarqué : Intel Atom

\*shelf\* pour les capteurs

Batterie :

### 4.1.4 Description Ordinateur Portable

#### HP ....

Processeur : Intel i5 ....

HD :

RAM :

### 4.1.5 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.

Description du capteur RGB-D : Asus Xtion Pro Live Résolution du capteur infrarouge Résolution de l'image : Range de vision : \*\* degrés :

1 2 3

## 4.2 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.

1. limitation de 5 mètres des capteurs infra-rouges.
2. Les objets qui touchent les bords.
3. illumination solaire, par exemple



#### 4.2.1 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>

### 4.2.2 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 existence 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 required 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.

L'Universidad de León a fait un compte rendu des *features* implementés sur PCL dans le lien \*[8]\*. Plus d'information sur les descripteur et ses implementations sur la librairie PCL peuvent être retrouvés sur le site internet

<http://pointclouds.org>.

Les images sont sauvegardées à l'aide de la librairie OpenCv dans le format .png (Portable Network Graphics).

Les nuages de points sont sauvegardées dans le format .pcd (Point Cloud Data). Pour sauvegarder les information \*provenientes\* de la segmentation, les [it]topics de sorti sont souscrit avec rosbag pendant le déplacement du robot. Les messages sauvegardés sont les suivants :

- v\_objects\_clouds : Vector de nuages de points obtenues pour chaque objet
- v\_objects\_image\_and\_mask : Sousimages et mask de chaque objet
- v\_object\_

## 4.3 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.