

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

Remerciements

Je remercie énormément Celine Craye pour son engagement et aide constant
pendant tout le déroulement de mon stage.
Merci.

Table des matières

1	Introduction	6
1.1	Contexte	6
1.2	Objectifs	6
1.3	Etat de l'art	6
2	Méthode proposée	8
2.1	Architecture générale	8
2.2	Segmentation	9
2.2.1	Algorithme	9
2.2.2	Restrictions	10
2.3	Descripteurs	10
2.4	Reconnaissance mono-vue	10
2.5	Localisation et suivi d'objet	11
2.5.1	Définition de repères	11
2.5.2	Bases mobiles	12
2.5.3	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	15
3	Protocole Expérimental	16
3.1	Matériel utilisé	16
3.2	Setup expérimental	16
3.3	Résultats expérimentaux	16
3.3.1	Comparaison à la reconnaissance mono-vue	16
3.3.2	Suivi et reconnaissance multi-cibles	18
4	Conclusion	21
4.1	Synthèse	21
4.2	Discussion	21
4.3	Perspectives	21
	Bibliographie	22
A	Matériels	24
A.1	Plateforme mobile	24
A.2	Ordinateur Portable	25
A.3	Capteur RGB-D	25
B	Logiciels	26

C Problèmes rencontrés	27
C.1 Synchronisation	27
C.2 Problèmes de déplacement	27
C.3 Restrictions logicielles	27
D Reconnaissance Mono-vue	28
D.1 Paramètres Segmentation	28
D.2 Descripteurs	28
D.2.1 Point Feature Histogram - PFH	28
D.2.2 Fast Point Feature Histogram - FPFH	29
D.2.3 Viewpoint Feature Histogram- VFH	29
D.2.4 Clustered Viewpoint Feature Histogram - CVFH	29
D.2.5 Estimation de la normale	30
D.2.6 Déplacement du robot	30
E Annexe II	31
E.1 Le groupe Thales	31
E.2 Secteurs d'activité	31
E.3 Présentation de ThereSIS	32
E.4 Secteurs d'activité	32
E.5 Le laboratoire Video Technologies & New Sensors	33

Introduction

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. 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, pendant que d'autres s'intéressent à l'utilisation du système de reconnaissance pour des tâches de recherche dans l'environnement [19, 25] et la manipulation d'objets [9, 13].

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

La majorité de la littérature traite le problème de la reconnaissance d'objets basés sur une seule image. Typiquement, un ensemble de *features* [3, 4, 18, 26] 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 [20], où l'image d'entrée est associée directement avec des classes des objets correspondants, au prix d'une étape d'entraînement importante d'apprentissage. De très nombreux exemples d'applications de la reconnaissance mono-vue existent dans le domaine de la robotique, pour la navigation sémantique [12], couplé avec l'estimation de pose pour la saisie de l'objet [11] ou encore pour la recherche d'objets dans l'environnement [19, 5].

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 [1]. 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 les limitation de la reconnaissance mono-vue [10]. Certaines approches se basent sur des modèles CAD à trois dimension, la description de contours et les graphes pour augmenter leur représentativité, une revue littéraire des approches est fait en [23]. En particulier, les graphes d'aspect [22] 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.

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 études comme [21] où traitement markovien fusionne les informations odométriques avec la reconnaissance d'imagez labellisées géographiquement pour la navigation d'un robot en environnement urbaine, travaillent sur le domaine multi-vues en y intégrant des aspects géométriques et ainsi augmenter la qualité de son estimation.

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 repositionner le capteur. Cela peut se faire par des critères de réduction de l'entropie en utilisant des probabilités de reconnaissance antérieures [8], en utilisant l'apprentissage par renforcement [7] où encore par estimation des faces cachées de l'objet [6]. Finalement, [9] traite ce même problème comme un problème de localisation et suivi par un filtre particulaire.

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 propriocep-tifs odométriques et d'une caméra RGB-D. Les informations provenant de 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.

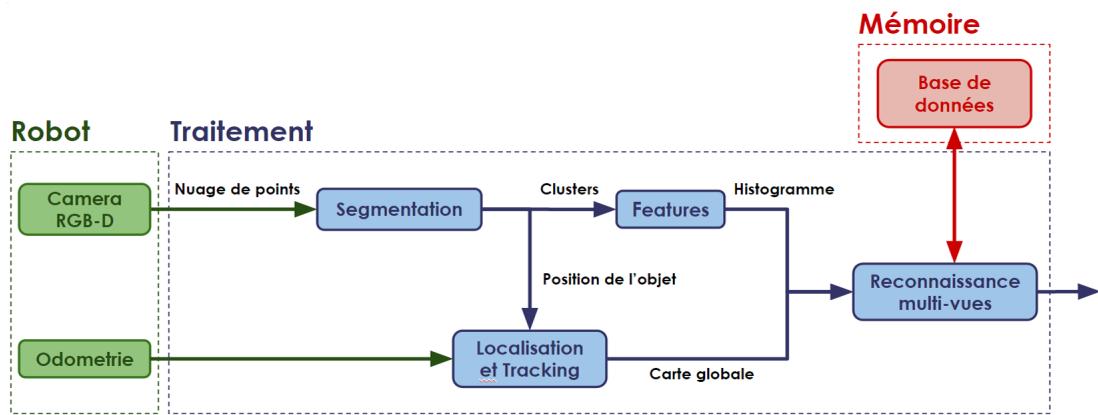


FIGURE 2.1 – Architecture générale du système

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.

Dans le cas où le capteur est immobile, on utilise classiquement des méthodes de soustraction de fond pour l'étape de segmentation [14]. 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. Parmi les méthodes de segmentation se basant sur cette définition, on peut citer Tabletop object detector [15] qui détermine le plan principal de l'image (généralement une table ou le sol) grâce à l'algorithme RANSAC [17], puis recherche des objets dans l'enveloppe convexe de ce plan. Par ailleurs, Caron et al. [12] 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 Algorithme

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 Theresis.

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.
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 grands plans orthogonaux au sol
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 associés.

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.

2.2.2 Restrictions

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 :

- Le sol où le robot se déplace n'est pas accidenté.
- L'objet se trouve par terre à une distance inférieure à 3 mètres .
- La lumière ambiante ne doit pas contenir trop de lumière infra-rouge.
- L'objet n'est ni transparent ni trop réfléctif 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. Une exemple de segmentation est présenté dans la figure 2.2 pour illustrer la capacité de segmentation.

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 FPFH [24] qui est invariant par changement de point de vue, SHOT [26] étant un descripteur local de courbure et des descripteurs semi-globaux orientés au traitement des occlusions CVFH [4] et Our-CVFH [3]. Une description détaillée de ces descripteurs et leurs principales différences sont expliquées dans les annexes D.2. 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

Afin de reconnaître les objets et leur points de vue rencontrés par le robot, nous utilisons les histogrammes de descripteurs VFH et une base de donnée réalisée à l'avance. Dans cette base, les histogrammes de plusieurs objets sont calculés pour plusieurs points de vues ainsi que leurs positions relatives (Plus de détails sur la construction de la base à la section 3.2). Le but est de retrouver l'objet et son point de vue le plus proche par rapport à la base. Pour cela, il est possible d'utiliser des algorithmes de *machine learning* classiques, mais les résultats sur des réseaux de neurones

n'ont pas été très concluants. Aldoma et al. [2] suggèrent l'utilisation de la mesure de similarité entre histogrammes chi-squared, associée au classificateur *k plus proches voisins*, ou K-NN. Le gros avantage de ce classificateur est l'étape d'apprentissage, qui correspond à création d'un arbre de recherche construit à partir de la comparaison croisée entre les éléments de la base. Par rapport aux données dont nous disposons, cet arbre se construit et fournit une estimation du plus proche voisin de manière presque instantanée.

L'API de la librairie FLANN sur PCL permet l'utilisation directe du classificateur K-NN. L'implémentation permet l'utilisation de plusieurs définitions de distance entre histogrammes. La définition par défaut, Chi-squared, dont la formule est donnée à l'équation 2.1, semble être capable de bien différencier les histogrammes d'entrés, H_1 et H_2 , et a été choisie pour notre système.

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \quad (2.1)$$



FIGURE 2.2 – **reconnaissance mono-vue** - Le résultat de la classification sur les objets segmentés où une écran et un ventilateur étaient reconnus. En rouge le plan du sol et en blanc les points à plus de 3 mètre considérés comme plus bruyants. Un remarque pour les ombres infra-rouges qui occultent les objets

2.5 Localisation et suivi d'objet

2.5.1 Définition de repères

Se placer dans différents repères permet d'avoir des référentiels plus naturels pour chaque type de composant du robot et pour les objets placés dans la scène. On définit quelques repères et conventions de base pour faciliter la localisation. D'abord le repère de la base du robot est orthonormale positif, où le déplacement vers l'avant correspond à l'axe \vec{x} , vers la gauche à l'axe \vec{y} et vers le haut à l'axe \vec{z} . Un deuxième référentiel utilisant les mêmes conventions 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 \vec{x} orienté vers la droite, l'axe \vec{y} vers le bas et enfin l'axe \vec{z} vers l'avant. Ces trois repères permettent d'orienter tous les éléments aperçus par le robot dans l'environnement de façon pratique.

Une méthode de transformation entre repères permet ensuite le passage de l'un à l'autre. On peut ainsi obtenir la position de l'objet dans le repère global d'après sa détection par la caméra. La transformation entre une base a et une autre b est faite par une matrice de transformation classique, décrit par l'équation 2.2.

$$\mathbf{R}_b^a = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta x \\ \sin \theta & \cos \theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

où θ équivaut à l'angle entre les deux repères et Δx et Δy sont les translations linéaires entre eux.

2.5.2 Bases mobiles

Estimation de l'odométrie

Certains robots sont dotés de capteurs capables d'estimer de façon approximative son déplacement. C'est aussi le cas du robot utilisé qui possède des roues codeuses capables d'estimer la rotation angulaire des roues. Pour le cas d'un robot différentiel, où chaque roue peut être commandée indépendamment, le déplacement et l'orientation suit les équations suivantes :

$$\begin{aligned} \delta x_t &= \delta s_t \cdot \cos(\theta_{t-1}) \\ \delta y_t &= \delta s_t \cdot \sin(\theta_{t-1}) \\ \delta \theta_t &= \frac{\omega_g + \omega_d}{d_r} \\ \delta s_t &= \frac{\omega_g + \omega_d}{2} \end{aligned} \quad (2.3)$$

ω_g et ω_d sont des respectives variations angulaire des roues et d_r est la distance entre elles. Une intégration, au sens mathématique 2.4, de la différence entre l'odométrie entre deux intervalles de temps permet de retrouver la position global du robot.

$$\begin{aligned} x_t &= x_{t-1} + \delta x_t \times \cos(\theta_{t-1}) - \delta y_t \times \sin(\theta_{t-1}) \\ y_t &= y_{t-1} + \delta x_t \times \sin(\theta_{t-1}) + \delta y_t \times \cos(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} + \delta \theta_t \end{aligned} \quad (2.4)$$

2.5.3 Filtre de Kalman

Afin de pouvoir utiliser le déplacement du robot par rapport aux objets pour les identifier, il est d'abord nécessaire de les localiser et les suivre. À cause de la divergence de l'odométrie, l'imprécision de la segmentation et le calcul du centroïde de l'objet, la position estimée est fortement bruitée et rend la suivie et identification infaisable lorsque plusieurs objets sont trop proches. Nous utilisons donc un filtre de Kalman pour corriger cette erreur de mesure et fournir une estimation plus fiable de la position des objets.

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

Prédiction

Une première de prédiction qui utilise le modèle linéaire \mathbf{F}_k pour décrire l'évolution des états au long du temps avec son bruit de process, \mathbf{Q}_k , associé et qui estime *a priori* la covariance de l'erreur $\mathbf{P}_{k|k-1}$. Formellement, on utilise les équations 2.5

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \quad (2.5)$$

Où les variables sont :

\mathbf{F}_k : la matrice de dynamique du système définie comme identité dans notre cas, si l'on considère que l'objet reste immobile

\mathbf{u}_k : l'entrée de commande, nulle dans notre cas

\mathbf{B}_k : la matrice qui relie l'entrée de commande u à l'état x , nulle également

$\mathbf{P}_{k|k-1}$: la matrice d'estimation a priori de la covariance de l'erreur

\mathbf{Q}_k : la matrice de covariance du bruit de process, diagonale dans notre cas.

Avec :

\mathbf{z}_k : observation ou mesure du process à l'instant k

\mathbf{H}_k : matrice qui relie l'état \mathbf{x}_k à la mesure \mathbf{z}_k

$\mathbf{P}_{k|k}$: matrice d'estimation a posteriori de la covariance de l'erreur

\mathbf{R}_k : matrice de covariance du bruit de mesure

Innovation

Une deuxième mise à jour, où l'observation est incorporée dans le calcul de l'innovation, $\tilde{\mathbf{y}}_k$, et du gain de Kalman, \mathbf{K}_k est décrite par l'équation 2.6.

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}\end{aligned}\tag{2.6}$$

Avec :

\mathbf{z}_k : l'observation s'agit de la position de l'objet segmenté dans le repère absolu

\mathbf{H}_k : la matrice qui relie l'état \mathbf{x}_k à la mesure \mathbf{z}_k : Ici, il s'agit d'une matrice identité puisque tout est réalisé dans le repère absolu.

$\mathbf{P}_{k|k}$: la matrice d'estimation *a posteriori* de la covariance de l'erreur

\mathbf{R}_k : la matrice de covariance du bruit de mesure. Matrice diagonale dans notre cas.

Suivi multi-cibles

Le caractère monomodal du filtre de Kalman nous contraint à ne pouvoir suivre qu'un seul objet à la fois. Pour obtenir un suivi multimodal, il faut que plusieurs filtres tournent en parallèle. Ainsi, le problème passe d'estimer la position d'un seul objet à celui de décider quelle observation appartient à quel filtre. Pour ce faire, nous définissons une matrice de distances entre chaque nouvelle observation et les états courants de chaque filtre de Kalman déjà créé. Ensuite, les nouvelles observations sont utilisées pour mettre à jour les filtres de Kalman dont l'estimation est la plus proche selon cette matrice. Avant toute mise à jour, on vérifie que la distance entre l'observation et l'estimation du filtre ne dépasse pas un certain seuil. Pour chaque observation qui n'a pas pu être associée à un filtre déjà existant, on crée alors un nouveau filtre.

2.6 Reconnaissance Multi-vue

2.6.1 Chaînes de Markov Cachées

Le déplacement physique du robot produit une séquence d'observations, sous différents points de vues, d'un même objet. On exploite l'information odométrique entre les différentes vues pour renforcer l'estimation de la vue d'un objet. De cette manière, l'évolution de la reconnaissance au cours du temps est représentée par un processus stochastique, dont une modélisation possible consiste 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 (c'est-à-dire en respectant la propriété de Markov de premier ordre), le processus stochastique est modélisée par une chaîne de Markov cachée.

Concrètement, les états cachés correspondent à des vues d'objets connus au préalable et déjà stockés dans la mémoire du robot. Cela constraint le nombre d'états et garantie que la chaîne soit finie. Puis, une matrice de transition, $a_{i,j}$, décrit l'évolution du processus. C'est cette matrice de

transition qui permet de prendre en compte l'odométrie et la transition entre les vues d'un même objets. Enfin, une autre matrice, $P(y_1 | k)$, dite matrice d'émission, estime la vraisemblance entre l'observation et les états de la chaîne.

Plus précisément, $a_{i,j}$ est définie en fonction de l'angle δ_{angle} , calculé par 2.7 qu'a parcouru le robot par rapport à l'objet entre deux vues successives. Dans notre modèle, on considère que $a_{i,j}$ est nulle si i et j sont deux vues d'objets différents. Pour deux vues i et j d'un même objet, séparées d'une distance d , le poids accordé à $a_{i,j}$ sera d'autant plus fort que δ_{angle} et d sont proches. D'autre part, la matrice d'émission $P(y_1 | k)$ correspond à la similarité entre l'histogramme d'un objet segmenté y et celui d'une vue d'objet dans la base de données k , normalisés par l'équation 2.8. La similarité est calculée comme l'inverse de la distance Chi square définie à la section 2.4.

$$\begin{aligned}\vec{d}_0 &= p_0 - p_{obj} \\ \vec{d}_1 &= p_1 - p_{obj} \\ \delta_{angle} &= \text{atan}(\vec{d}_1) - \text{atan}(\vec{d}_0)\end{aligned}\tag{2.7}$$

La transformation des distances des histogrammes en probabilité est faite d'après la normalisation suivant :

$$\mathbb{P}(y|x, database) = \frac{\sum_a d_a^x - d_y^x}{\sum_b \sum_c d_c^x - d_b^x}\tag{2.8}$$

où x est l'image de teste et y un élément de la base de donné. Dans le cas du plus proches voisin la normalisation prend en compte seulement les k plus proches histogrammes, en contraste au approche *bruta force*.

Une autre modélisation possible aurait été d'avoir une chaîne de Markov cachée distincte pour chaque objet et ensuite décider à chaque pas de temps le processus le plus vraisemblable. Cette modélisation peut être vue comme un sous-ensemble du cas précédent où les transitions entre deux objets ne sont pas considérées. Pourtant, il peut arriver que deux objets soit considérés comme positionnés au même endroits, ou bien que des objets mobiles fusionnent (par exemple, une personne qui viendrait s'asseoir sur une chaise, ou encore un personne qui commence à marcher)¹.

2.6.2 Algorithme de Viterbi

Reste donc à extraire des informations de la modélisation Markovienne proposée. La séquence d'états la plus vraisemblable qui pourrait avoir généré les observations y_1, \dots, y_T , correspond normalement à la séquence d'objets reconnus. Afin de retrouver cette séquence, aussi appellée chemin, on fait appel à la programmation dynamique, et plus spécifiquement à l'algorithme de Viterbi, d'où le nom chemin de Viterbi. L'algorithme retrouve de façon récursive l'état courant le plus probable, en prenant en compte seulement les observations jusqu'à un instant donné et son estimation aux instants antérieurs. Ceci se traduit par les équations 2.9

$$\begin{aligned}V_{1,k} &= P(y_1 | k) \cdot \pi_k \\ V_{t,k} &= \max_{x \in S} (P(y_t | k) \cdot a_{x,k} \cdot V_{t-1,x})\end{aligned}\tag{2.9}$$

Ici, $V_{t,k}$ représente la probabilité que la séquence d'états la plus probable finisse dans l'état k , ayant généré les observation à l'instant t , tandis que π_i représente la probabilité initiale de se retrouver en chaque état. Pour retrouver le chemin de Viterbi, il suffit de trouver le maximum de $V_{t,k}$:

$$x_T = \arg \max_{x \in S} (V_{T,x})\tag{2.10}$$

1. Le fait de se mettre en mouvement altère les formes d'une personne, ce qui rend possible sa détection comme un nouvel objet.

2.6.3 Graphe d'aspect polaire

On considère que les objets sont décrits par deux dimensions d'information : une spatiale, représentant la position absolue de l'objet dans l'environnement ainsi que les positions relatives où l'objet a été visualisé, et une autre dimension visuelle, donnée par les descripteurs géométriques, de couleurs et de texture. On cherche à représenter cette description dans un référentiel unique. Le graphe d'aspect permet de coupler l'ensemble des 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 - le nombre d'images intégrées au modèle.

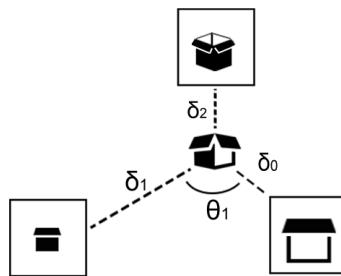


FIGURE 2.3 – Représentation des objets par un modèle polaire

Un référentiel polaire entrelace toutes ces informations de façon à représenter la position spatiale d'où l'observation a été faite, comme représenté dans l'image 2.3. Pour la construction du modèle les conventions suivantes ont été 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 des features visuelles ne sont pas invariantes à l'échelle, et ce d'autant plus si la résolution de l'image joue un rôle critique pour la détection de features, comme les patches SIFTs. Ainsi, prendre également en compte la distance à laquelle l'image a été prise peut être intéressant pour limiter la classification à une échelle valable.

Protocole Expérimental

3.1 Matériel utilisé

En ce qui 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 ROS¹ a été adopté pour avoir les bibliothèques qui permettent de gérer les nuages de points, Freenect, OpenNi2 et PCL², 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

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

1. Robot Operating System
2. Point Cloud Library

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 ± 3 seconds.

La difficulté de l'évaluation vient, premièrement, du fait que la base de donnée a été réalisée avec très peu de vues, ce qui donne lieu à des mauvaise reconnaissance mono-vue lorsque le point de vue observé se situe entre deux vues de la base de connaissance. De plus, la vitesse de déplacement peut générer des images plus floues lors des acquisitions et la modification de l'angle de la caméra³ apportent un obstacle en plus pour le *matching* de descripteurs dans le classificateur K-NN.

Un expérience typique est illustrée dans l'image 3.2 :

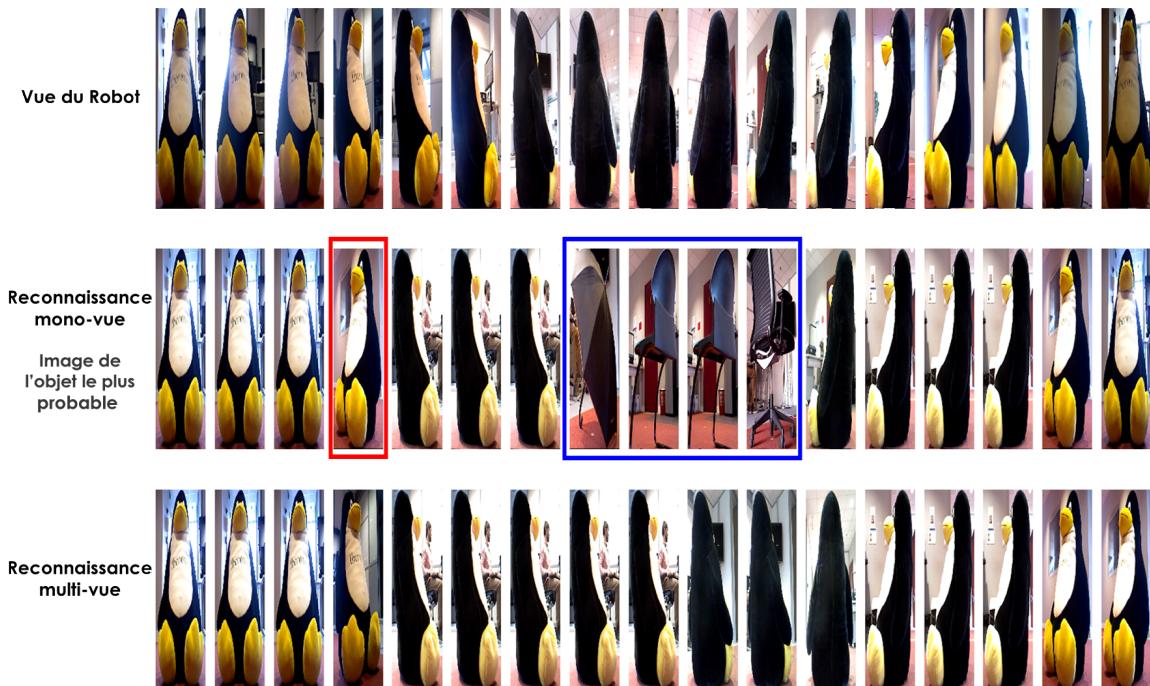


FIGURE 3.2 – **Expérience typique** - Reconnaissance multi-vue corrige des ambiguïtés et surmonte la mauvaise segmentation lors de la création de la base.

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.

Le résultat de l'évaluation peut être présenté en forme d'une courbe : nombre de visualisations de l'objet dans les abscisses, par rapport aux taux de reconnaissance d'objet pour chaque système de reconnaissance dans l'estimation de l'objet et de l'orientation.

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

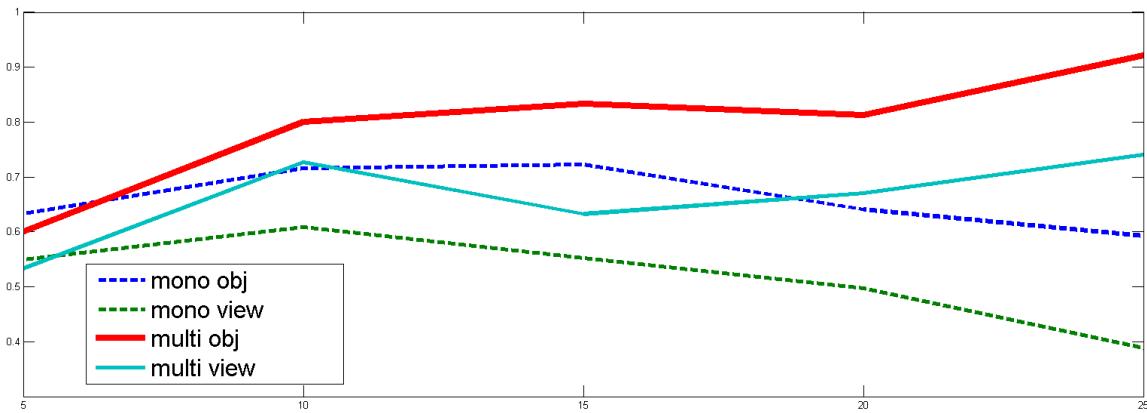


FIGURE 3.3 – **Résultat de l'évaluation** - Les courbes pointillés représentent le système d'après une seule image fixe (mono-vue), pendant que les complètes correspondent au système multi-vue. Avec un nombre réduit d'images la reconnaissance mono-vue tend à être plus performante . Avec quelques vues différents la taux de reconnaissance commence à monter ayant un écart de 33 % après le tour complet avec un valeur absolu de 92% de réussit pour l'estimation de l'objet et 74 % pour l'estimation de l'orientation.

La courbe de la figure 3.3 permet conclure, d'abord, que l'algorithme de reconnaissance multi-vue est plus performant que sa correspondant mono-vue lorsque le nombre d'observations augmentes⁴. Les haut taux de reconnaissance dans En deuxième temps que l'estimation de l'orientation de l'objet est plus difficile que sa reconnaissance, renforcé par le taux inférieur de positifs pour les deux systèmes

3.3.2 Suivi et reconnaissance multi-cibles

La deuxième expérimentent correspond à placer des objets présents dans la base de données dans une pièce et conduire le robot en faisant en sorte qu'il les regarde sous plusieurs points de vues différents. Ce scénario est beaucoup plus complexe que celui d'avant. D'abord les objets occultent uns aux autres, donnant lieux à mauvaises segmentations. Ensuite, le suivi des objets est beaucoup plus complexe où objets proches peuvent être confondus.

La carte finale donnée par l'algorithme est affiché en 3.4. Une photo de la pièce aves les objets disposés comme dans l'expériment peut être retrouvé en 3.5.

4. Ce fait peut être associé à la démarrage du robot où des images était relativement constants grâce à une vitesse de déplacement moins important au début des séquences.

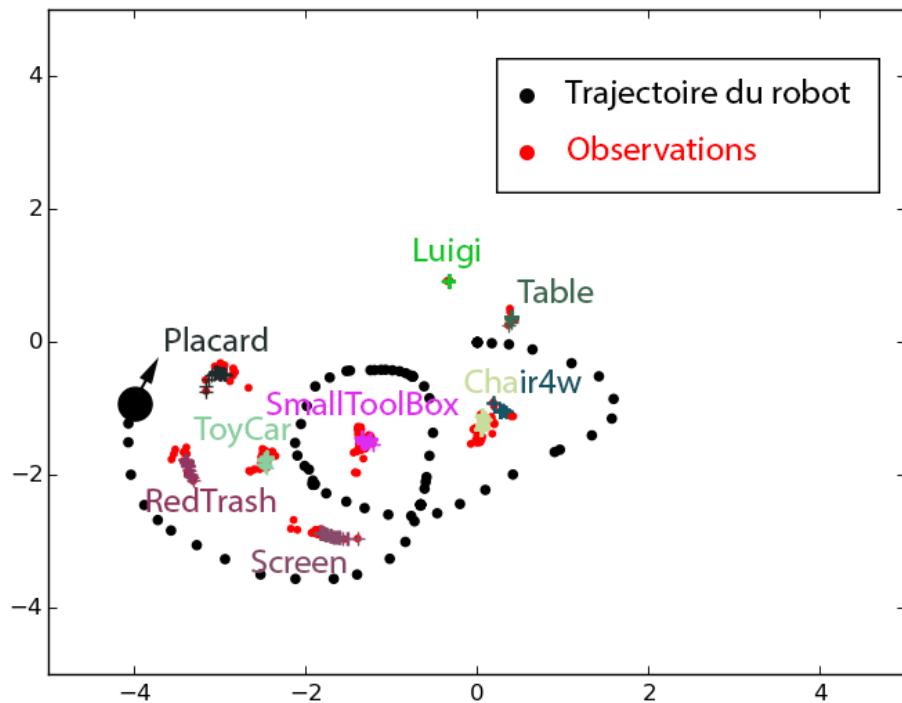


FIGURE 3.4 – Résultat du déplacement du robot - Les boules rouges correspondent à les observations des postions des objets, chacun représenté par une croix de couleur différent dans la scène donné par l'algorithme de suivi multi-cibles. En noir la trajectoire du robot bien comme sa dernière position.



FIGURE 3.5 – Un des placement des objets choisi pour le deuxième expriment



FIGURE 3.6 – Reconnaissance Multi-cible - Quatre entre les cinq objets présentes dans la scène ont été reconnus bien reconnu avec une estimation d'orientation raisonnable. Le première objet (personne) était mal segmenté ensemble avec le ventilateur.

Les résultat de l'expériment montre que le système est capable de gérer ce cas beaucoup plus complexe, pourtant avec une réduction des taux de reconnaissance. Quelques améliorations proposés dans la section suivante pourrait être utiles pour une meilleur sa performance. Un exemple de la reconnaissance multi-cible est mis dans les annexes.

Conclusion

4.1 Synthèse

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). Elle se montre plus performant quand comparé à son correspondent fixe d'après une première évaluation, ayant un taux de réussite de 92% lorsqu'un tour complète est fait pour la reconnaissance d'objet, en contraste aux 59% du système classique. Pour le défi de reconnaître l'orientation spatial le taux diminue mais restant encore assez élevé autour de 75%.

4.2 Discussion

Même ayant des résultats intéressants quelques limitations apparaissent lorsqu'on se déplace dans environnement plus exigeants et qu'on veut se déplacer librement. Regarder les objets de trop près, par exemple, coupe une partie de l'image qui résulte dans un mauvais classification de l'objet ce qui restreint les zones de déplacement. De même objets placés à une distance inférieur à 1 mètre entre eux risquent d'être mélangés par le tracking. Le modèle est aussi sensible à la densité de la base de donné, car avoir une base trop discrète résulte en points de vues inexistant qui sont souvent associés à d'autres objets.

4.3 Perspectives

La formulation de la chaîne de Markov cachée est suffisamment général pour incorporer des nuances plus complexes en cas d'occlusion, de changement ou d'association d'objet (humain+chaise). Explorer ce potentiel semble agrandir encore plus la puissance du modèle. Au même temps, la communication entre la reconnaissance, qui suggérait des modèles cinématiques de déplacement, et le module de suivi multi-cible peut faire en sorte qu'environnements plus complexes avec des objets mobiles où la position physique des objets, des fois, n'est pas suffisant pour les déterminer puissent être gérés. Une amélioration un peu plus immédiate serait d'ajouter des features de couleur et de texture pour lever les ambiguïtés de vues. Finalement intégrer un algorithme de SLAM pour rendre plus robustes les estimations de position et avoir une meilleure représentation de l'environnement couplé avec 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.

Bibliographie

- [1] Alaa E Abdel-Hakim, Aly Farag, et al. Csift : A sift descriptor with color invariant characteristics. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1978–1983. IEEE, 2006.
- [2] A. Aldoma, Zoltan-Csaba Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R.B. Rusu, S. Gedikli, and M. Vincze. Tutorial : Point cloud library : Three-dimensional object recognition and 6 dof pose estimation. *Robotics Automation Magazine, IEEE*, 19(3) :80–91, Sept 2012.
- [3] Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu, and Markus Vincze. *OUR-CVFH-Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation*. Springer, 2012.
- [4] Aitor Aldoma, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, and Gary Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 585–592. IEEE, 2011.
- [5] Haider Ali, Faisal Shafait, Eirini Giannakidou, Athena Vakali, Nadia Figueroa, Theodoros Varvadoukas, and Nikolaos Mavridis. Contextual object category recognition for rgbd scene labeling. *Robotics and Autonomous Systems*, 62(2) :241–256, 2014.
- [6] Joseph E Banta, Laurana M Wong, Christophe Dumont, Mongi Abidi, et al. A next-best-view system for autonomous 3-d object reconstruction. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 30(5) :589–598, 2000.
- [7] 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.
- [8] Hermann Borotschnig, Lucas Paletta, Manfred Prantl, Axel Pinz, et al. Active object recognition in parametric eigenspace. In *BMVC*, pages 1–10. Citeseer, 1998.
- [9] Björn Browatzki, Vadim Tikhanoff, Giorgio Metta, Heinrich H Bülthoff, 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.
- [10] Heinrich H Bülthoff, Christian Wallraven, and Arnulf Graf. View-based dynamic object recognition based on human perception. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 768–776. IEEE, 2002.
- [11] Gilles Burel and Hugues Hénocq. Three-dimensional invariants and their application to object recognition. *Signal Processing*, 45(1) :1–22, 1995.
- [12] 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.

- [13] Alvaro Collet and Siddhartha S Srinivasa. Efficient multi-view object recognition and full pose estimation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2050–2055. IEEE, 2010.
- [14] Ke-xue DAI, Guo-hui LI, Dan Tu, and Jian YUAN. Prospects and current studies on background subtraction techniques for moving objects detection from surveillance video. *Journal of Image and Graphics*, 11(7) :919–927, 2006.
- [15] Ros documentation. Tabletop object detector.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] Danica Kragic. Object search and localization for an indoor mobile robot. *CIT. Journal of Computing and Information Technology*, 17(1) :67–80, 2009.
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [21] 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.
- [22] 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.
- [23] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Active recognition through next view planning : a survey. *Pattern Recognition*, 37(3) :429–446, 2004.
- [24] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.
- [25] Ksenia Shubina and John K Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5) :535–547, 2010.
- [26] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Computer Vision–ECCV 2010*, pages 356–369. Springer, 2010.

Matériels

A.1 Plateforme mobile

Robot Wifibot v2

Dimensions :

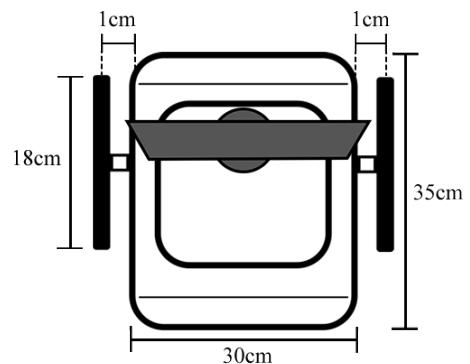
Hauteur : 18 cm

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



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 librairies d'acquisition d'images RGB-D, OpenNi 2 et Freenect, ainsi qu'une librairie 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.

<https://github.com/PointCloudLibrary/pcl/wiki/>

<http://pointclouds.org/documentation/tutorials/>

Problèmes rencontrés

C.1 Synchronisation

La synchronisation entre tous les modules du robot s'est montrée d'extrême importance pour le bon fonctionnement de du système. Retards trop importants résultent en mauvaises transformations de repère, par conséquent, le système de suivi multi-cible n'arrive pas à bien distinguer des objets trop proches.

C.2 Problèmes de déplacement

La roulette du 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 crée un mouvement parasite qui perturbe la trajectoire voulue. Nous avons tenté sans succès d'installer une bille omnidirectionnelle à roulement, qui se bloquait sur la moquette avec le poids du robot. Une deuxième solution serait d'interdire certains mouvements du robot pour éviter cette déviation.

C.3 Restrictions logicielles

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

Reconnaissance Mono-vue

D.1 Paramètres Segmentation

Le méthode de segmentation exige une définition a priori des paramètres pour le bon fonctionnement du système.

- Taille du *grid* de voxalization : 2 cm
- Distance maximale au capteur : 3 m
- Rayon d'estimation de la normale : 2 cm
- Aire de *smoothing* de la normale : 10 cm^2
- Distance pour qu'un point soit considéré comme appartenant au plan : 5 cm
- Distance minimale du plan du sol pour qu'il soit considéré comme partie de l'objet : 3 cm

La pluparts des valeurs ont été choisies telle qu'elles étaient proposées dans la librarie PCL. Certaines ont été modifiées pour atteindre les caractéristiques attendues.

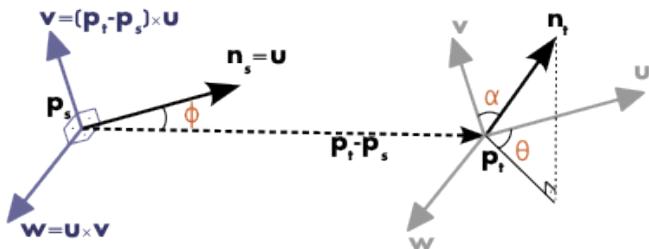
D.2 Descripteurs

D.2.1 Point Feature Histogram - PFH

Le PFH utilise 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, dont certains seront expliqués par la suite.

En revenant à son calcul, l'histogramme est évalué à partir des paires de points à l'intérieur d'un ensemble prédéfini. D'abord, un repère initial, illustré dans l'image ?? est établi sachant le vecteur distance normalisé et les normales aux deux points. 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 considérées comme features pour réduire l'espace initial à douze dimensions - coordonnées et normales des deux point - à un espace à quatre dimensions.

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



Puis, les normales sont transformées en features angulaires décrit par les équations D.1

$$\alpha = \mathbf{v} \cdot \mathbf{n}_t \quad \phi = \mathbf{u} \cdot \frac{(\mathbf{p}_t - \mathbf{p}_s)}{d} \quad \theta = \arctan(\mathbf{w} \cdot \mathbf{n}_t, \mathbf{u} \cdot \mathbf{n}_t) \quad d = \|\mathbf{p}_t - \mathbf{p}_s\|_2 \quad (\text{D.1})$$

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

Le PFH est robuste à des différents échelles de densité de points et de bruit, mais aussi invariant à des transformations affines. Ses inconvénients sont liés à la dépendance de la qualité de l'estimation de la normale.

D.2.2 Fast Point Feature Histogram - FPFH

L'utilisation du FPFH vient de la volonté de réduire la complexité du calcul du descripteur PFH, $O(nk^2)$, pour un nuage avec n points où chacun des points a k voisins. Pour cela, l'algorithme, au lieu de calculer la relation bidirectionnelle entre tous deux points de l'ensemble définis, pondère les features de chaque point par les voisins à l'intérieur d'un rayon de recherche, selon la formule D.2

$$FPFH(\mathbf{p}_q) = SPFH(\mathbf{p}_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_k} \cdot SPFH(\mathbf{p}_k) \quad (\text{D.2})$$

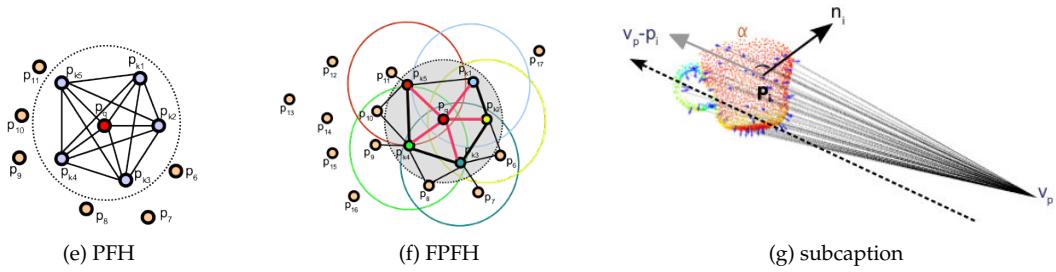
Cette procédure a maintenant une complexité $O(n*k)$. Le gain en vitesse est donc considérable, ce qui lui permet d'être appliquée pour des utilisations en temps réel. De plus, pour éviter une perte d'information considérable, le FPFH incorpore quelques points externes au rayon de voisinage, mais qui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

D.2.3 Viewpoint Feature Histogram- VFH

Le VFH, contrairement à PFH et FPFH, est une extension du deuxième descripteur où la variance de point de vue est prise en compte. Succinctement, des angles entre la normale de chaque point et la direction principale d'observation sont concaténés à l'histogramme provenant du SPFH (Simplified PFH). En gardant le repère utilisé dans les descripteurs précédents, le vecteur de direction principale est défini par la différence entre l'origine du capteur jusqu'au centroïde du nuage de point considéré. Ce type de feature permet de reconnaître à la fois l'objet et son orientation spatiale. Par conséquent, c'est la feature utilisée dans notre système.

D.2.4 Clustered Viewpoint Feature Histogram - CVFH

CVFH - Clustered VFH - est une feature semi-globale capable de gérer des occlusions partielles, des erreurs de segmentation et du bruit. Ceci est possible grâce à 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 fort gradient de courbure, considérés comme zones de transitions entre surfaces. Puis, 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 d'histogrammes. En revanche, le découpage exige un soin plus important avec la résolution des surfaces afin qu'elles restent représentatives de l'objet.



L'Universidad de León a fait un compte rendu des *features* implémentés 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>. **n'oublie pas de mettre un lien**

D.2.5 Estimation de la normale

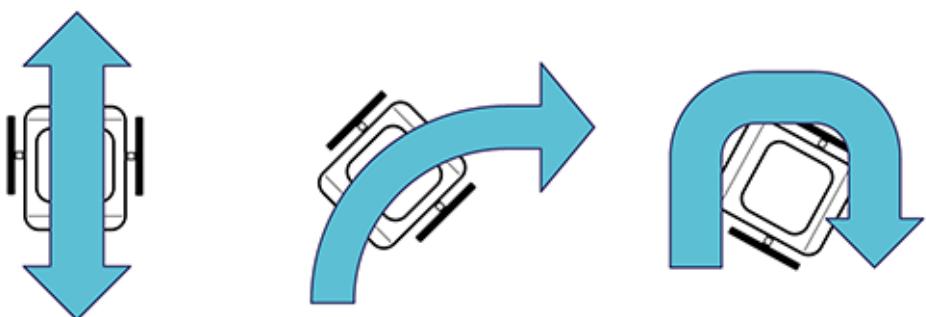
Pour constituer les informations géométriques l'estimation de la normale du point est d'extrême importance. Sont calcul est fait de la manière suivant :

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 bibliothèque 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.

D.2.6 Déplacement du robot

Le robot est équipé de trois roues, dont les deux roues symétriques arrières sont motorisées et responsables du déplacement moteur. D'autre part, la dernière sert à donner un support pour la partie arrière du châssis. Les moteurs sont contrôlés à partir de commandes serial, préétablis par 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 d'accéder à n'importe quelle position de l'espace.

Annexe II

E.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.

E.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.

E.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.

E.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.

E.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 CEA¹, baptisé VisionLab.

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