

# An Odometry-Based Reinforcement Approach for Indoor Objects and Viewpoint Recognition

Luigi F. Tedesco<sup>1</sup>, Céline Craye<sup>2</sup>, Jean-François Goudou<sup>3</sup> and David Filliat<sup>4</sup>

**Abstract**— Object recognition capability is a essential condition for giving autonomy to mobile robots in human made environment. However, achieving this goal by means of visually representing objects is a hard task ?? and using all possible sources of information is a must. Here we present a procedure to incorporate the notion of continuity and overcome ambiguous points of view. By observing objects from different perspectives binded with a Markovian modeling of the stochastic processes of recognizing each of the objects viewpoint, the algorithm copes with a sparse database, blurred images from motion and object spatial symmetry, to recognize and estimate objects 6-dof pose. A multi-modal Kalman based tracking was also implemented in order to recognize multiple objects simultaneously. The approach was tested in a mobile platform and the comparison between the single viewed and the proposed recognition gave promising results.

## I. INTRODUCTION

The vast majority of the literature focus on single image object visual recognition for helping robots in tasks such as semantic navigation ??, pose estimation for grasping ?? and environmental search ???. Typically, a set of features is extracted from a segmented object candidate and, subsequently, compared to a database of priori known objects. Extensive work have been done in order to increase efficiency in each one of the sub-processing steps. Among them : segmentations methods using range cameras, features that describe color and texture ??, geometry ??, contours ??, besides classifiers and matching techniques. Alternatively, a deep neural architecture ?? can perform a direct object visual classification after a delicate training phase. However, the classic recognition pipeline seems to be more natural and simple to be implemented with a straight-forward training, still having reasonable results.

Nevertheless, ambiguous viewpoints easily trick visual descriptors reducing its recognition capability. Observing objects sequentially from distinctive points of view seems to be a natural way to deal with the problem. A solution inspired by human behavior for learning new unseen objects has been proposed by ??, using key-frames and the rate of matching features with past frames, to overcome ambiguity in face recognition task. More work have been done to model objects different viewpoints perspectives summarized by Roy and al. ??.

<sup>1</sup>Luigi Franco Tedesco is an undergraduate student with Faculty of Electrical Engineering, Robotics and Artificial Intelligence, Thales Service, 91767 Palaiseau, France tedesco@ensta.fr

<sup>2</sup>Bernard D. Researcher is with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA b.d.researcher@ieee.org

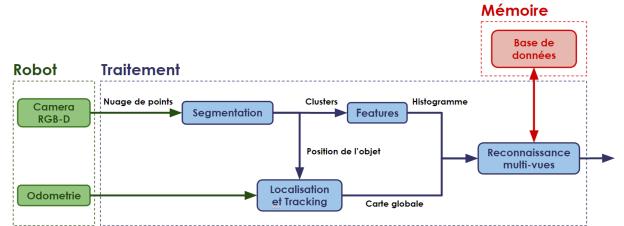


Fig. 1. General achitecture of the system

## II. PROPOSED APPROACH

### A. General architecture

Our approach was designed for robots equipped with odometric sensors and an RGB-D camera. Informations from these units are sent to a processing module that isolates objetc from input images extracts features that describe each objetc, and compared them with a reference database stored in memory. When no matching is found between observation and the database, a new object can be added to the database and increase the robot's knowledge about the environment.

The architecture of the system is illustrated in Figure 1 and explains the dependencies between the different processing modules and the input and output flow for each of them.

More precisely, the processsing units takes as an input the point cloud from the RGB-D sensor as well as the odometry measure of the two wheels. The first module consists in a segmentation step that cleans up the point cloud by isolationg objects of the scene. A new point cloud for each object is then created. Those point clouds are then sent to the feature extraction module to convert them to discriminative feature histograms. On the other hand, another module converts the odometry data to a localization and displacement of the robot in an absolute reference. Each segmented object can then be localize in the same reference and sent to the recognition module. The recognition module uses both the object feature and the evolution of those features compared to the displacement of the robot to retrieve the most likely object and view.

### B. Object Segmentation

The segmentation step aims to differentiate objects from the background of raw images. Stereoscopic and infra-red cameras helped the treatment adding a new dimension to images and allowing segmentation geometrically. In the case where the sensor is motionless, background subtraction approach are typically used for segmentaiton [5]. This is not

applicable in our case as the robot is constantly moving in its environment. By making the hypothesis that objects are represented in the scene by as cluster of points right above the ground plan, other approaches such as *Tabletop object detector* [6] determine the main plane of the image from depth data and search for elements in the convex hull of the plane that are not the plane itself. For indoor applications, this assumption is usually enough to isolates objects on tables or floor.

The segmentation algorithm used in this work is the one proposed by Caron et al. [4]. The approach uses the same assumption that objects are lying on planes surfaces that can be found as the major plane of images. In our case, as the camera on the robot is always keeping the same orientation towards the floor, the floor plane equation can be found during a calibration step and used during the whole experiment. The segmentation algorithm uses the following steps

start=0

- 1) Calibration to obtain the floor plane equation before experiments, using RANSAC algorithm on an image where floor is the main plane.

Then for each frame of the experiment:

- 1) Floor subtraction from the obtained equation
- 2) Points filtering if the distance to the sensor is more than 3 meters
- 3) Normal surfaces of the remaining points calculation
- 4) Filtering of big planes orthogonal to the ground. They are very likely to be walls
- 5) Voxelization of remaining points to speed up processing
- 6) Projection of those points on the floor plane
- 7) Clustering of the projected points to create object candidates
- 8) Elimination of clusters that are too close to borders. They might not be objects, and their missing parts are likely to badly influence the recognition.
- 9) Centroid calculation of each remaining cluster (should be the objects of the image)

Based on this algorithm, we obtain the position of each object in the camera frame, as well as the associated point cloud and normals.

### C. Feature

Among all kind of image features, the Viewpoint Feature Histogram captures the object geometry by estimating the angular transformation between the normal of each of the object's point and the standpoint from where it has been viewed. The interest of using such a feature is to explore the ambiguity created from objects spatial symmetry.

### D. Aspect-Graph

In order to represent objects in the 3 dimensional space, an aspect graph representation merges viewpoint appearance and the necessary movement to transit between them.

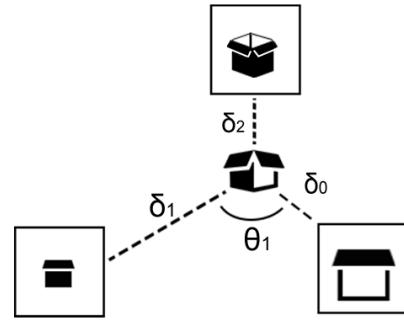


Fig. 2. Reprsentation des objets par un modle polaire

### E. Object representation

Among all kind of image features, the Viewpoint Feature Histogram captures the object geometry by estimating the angular transformation between the normal of each of the object's point and the standpoint from where it has been viewed. The interest of using such a feature is to explore the ambiguity created from objects spatial symmetry.

L'objectif de notre mthode est d'avoir une reconnaissance multi-vues d'un ou plusieurs objets la fois, capable d'intgrer le dplacement du robot pour rsoudre des ambiguts et faux positifs. Pour incorporer les notions de vues et de transition entre elles, on utilise une reprsentation simple et suffisamment gnrale base sur les graphes d'aspect. Le dplacement d'un tat un autre dans ce graphe est ensuite estim par rapport au dplacement du robot. Ce systme est ensuite coupl avec un dispositif de reconnaissance mono-vue classique capable de retrouver la vue la plus probable d'un objet partir de descripteurs 3D. Une mthode de suivi des objets et un traitement probabiliste de changement de vue tant donn l'information motrice permet enfin d'augmenter le taux de reconnaissance.

Afin de reconnatre les objets et leurs points de vue rencontrs par le robot, nous utilisons une base de donne ralise l'avance. Dans cette base, des descripteurs de plusieurs objets sont calculs pour plusieurs points de vues ainsi que leurs positions relatives (Plus de dtails sur la construction de la base la section ??). Afin de stocker et accder aux donnees des objets de la base en mmoire, nous utilisons une reprsentation base sur un graphe d'aspect polaire.

1) *Graphe d'aspect polaire:* On considre que les objets sont dcrits par deux dimensions d'information : une spatiale, reprsentant la position absolue de l'objet dans l'environnement ainsi que les positions relatives o l'objet a t visualis, et une autre dimension visuelle, donne par les descripteurs gomtriques, de couleurs et de texture. On cherche reprsenter cette description dans un rfrentiel unique. Le graphe d'aspect permet de coupler l'ensemble des images suivant ses possibles transitions spatiales, ce qui rsulte dans la possibilit de construire le modle la vole et de jouer avec sa densit d'information - le nombre d'images intgres au modle.

Un référentiel polaire permet d'intégrer toutes ces informations de façon à représenter la position spatiale d'où l'observation a été faite, comme représenté dans l'image 2. 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 centre 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.

*2) 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 aux transformations affines et aux changements de luminosité. On s'intéresse surtout ici aux descripteurs basés sur le nuage de points 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. Parmi les descripteurs 3D proposés dans la littérature, on peut citer FPFH [7] qui est invariant par changement de point de vue, SHOT [8] qui est un descripteur local de courbure et des descripteurs semi-globaux utilisés au traitement des occlusions, CVFH [3] et Our-CVFH [2]. Une description détaillée de ces descripteurs et leurs principales différences sont expliquées dans les annexes ???. Nous choisissons d'utiliser le descripteur *Viewpoint Feature Histogram - VFH*, car il permet de discriminer non seulement les formes géométriques (pour la reconnaissance d'objet), mais aussi les points de vues (reconnaissance de vue).

En partant du principe 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 segmenté, on obtient un histogramme VFH représentatif de l'objet et de sa vue courante. D'autre part, les histogrammes VFH de tous les éléments de la base de données sont calculés au préalable pour être comparés plus rapidement.

*3) Reconnaissance mono-vue:* 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 obtenus avec des réseaux de neurones n'ont pas toujours été concluants. Aldoma et al. [1] 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 la 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.

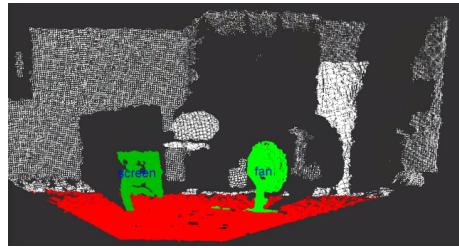


Fig. 3. **reconnaissance mono-vue** - Résultat de la classification à partir d'une seule vue sur les objets segments comprenant un crane et un ventilateur. En rouge, le plan du sol et en blanc les points plus de 3 mètres, non pris en compte dans la segmentation. On remarque également les ombres infrarouges qui occultent les objets

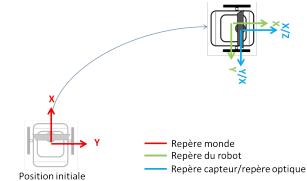


Fig. 4. Représentation des différents repères utilisés

L'API de la bibliothèque 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 1, semble être capable de bien différencier les histogrammes d'entrées,  $H_1$  et  $H_2$ , et a été choisie pour notre système.

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \quad (1)$$

#### F. Object localization and tracking

*1) Définition des 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. Tout d'abord, le repère de la base du robot est orthonormal 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  $\vec{z}$ . Le repère mondial (ou repère absolu) est choisi comme étant le repère du robot dans sa position initiale. Un troisième 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.

La figure 4 permet de visualiser les différents repères utilisés.

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 à partir de sa détection par la caméra. La transformation entre une base  $a$  et une autre  $b$  est faite par une matrice de transformation classique, décrite par l'équation 2.

$$\mathbf{R}_b^a = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta x \\ \sin \theta & \cos \theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

o  $\theta$  quivaut l'angle entre les deux repres et  $\Delta x$  et  $\Delta y$  sont les translations entre les deux origines.

### 2) Bases mobiles:

3) *Estimation de l'odomtrie:* Certains robots sont dts de capteurs capables d'estimer de faon approximative leurs dplacements. C'est aussi le cas du robot utilis qui possde des roues codeuses capables d'estimer la rotation angulaire des roues. Pour le cas d'un robot diffrentiel, o chaque roue peut tre commandé indpendamment, le dplacement et l'orientation suivent les quations suivantes:

$$\begin{aligned} \delta x_t &= \delta s_t \cdot \cos(\theta_{t-1}) \\ \delta y_t &= \delta s_t \cdot \sin(\theta_{t-1}) \\ \delta \theta_t &= \frac{\delta \omega_g + \delta \omega_d}{d_r} \\ \delta s_t &= \frac{\delta \omega_g + \delta \omega_d}{2} \end{aligned} \quad (3)$$

$\omega_g$  et  $\omega_d$  sont des respectives variations angulaire des roues droites et gauches, et  $d_r$  est la distance entre elles. Une intgration, au sens mathmatique, de l'odomtrie entre deux intervalles de temps permet de retrouver la position global du robot par l'quation 4:

$$\begin{aligned} x_t &= x_{t-1} + \delta x_t \times \cos(\theta_{t-1}) - \delta y_t \times \sin(\theta_{t-1}) \\ y_t &= y_{t-1} + \delta x_t \times \sin(\theta_{t-1}) + \delta y_t \times \cos(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} + \delta \theta_t \end{aligned} \quad (4)$$

4) *Filtre de Kalman :* Afin de pouvoir utiliser le dplacement du robot par rapport aux objets pour aider leur identification, il est d'abord ncessaire de les localiser et les suivre. cause de la divergence de l'odomtrie, l'imprcision de la segmentation et le calcul du centrode de l'objet, la position estime est fortement bruite et rend le suivie et identification infaisables lorsque les 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 jour selon deux tapes :

5) *Prdiction:* Une premire de prdition qui utilise un modle de dynamique linaire  $\mathbf{F}_k$  pour dcrire l'volution des tats au long du temps avec son bruit de process  $\mathbf{Q}_k$  associ, et qui estime *a priori* la covariance de l'erreur  $\mathbf{P}_{k|k-1}$ . Formellement, on utilise les quations 5

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \quad (5)$$

O les variables sont :

$\mathbf{F}_k$  : la matrice de dynamique du systme dfinie comme identit dans notre cas, si l'on considre que l'objet reste immobile

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

$\mathbf{B}_k$  : la matrice qui relie l'entre de commande  $u$  l'tat  $x$ , nulle galement

$\mathbf{P}_{k|k-1}$  : 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.

6) *Innovation:* Une deuxime mise jour, o l'observation est incorpore dans le calcul de l'innovation,  $\tilde{\mathbf{y}}_k$ , et du gain de Kalman,  $\mathbf{K}_k$  est dcrite par l'quation 6.

$$\begin{aligned} \tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \end{aligned} \quad (6)$$

Avec :

$\mathbf{z}_k$  : l'observation c'est--dire la position de l'objet segment dans le repre monde

$\mathbf{H}_k$  : la matrice qui relie l'tat  $\mathbf{x}_k$  la mesure  $\mathbf{z}_k$  : Ici, il s'agit d'une matrice identit puisque tout est reprsent dans le repre monde.

$\mathbf{P}_{k|k}$  : 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.

7) *Suivi multi-cibles:* Le caractre monomodal du filtre de Kalman nous contraint ne pouvoir suivre qu'un seul objet la fois. Pour obtenir un suivi multimodal, il faut que plusieurs filtres tournent en parallel. Ainsi, le problme passe destimer la position d'un seul objet celui de dcider quelle observation appartient quel filtre. Pour ce faire, nous dfinissions une matrice  $M_{i,j}$  de distances entre chaque nouvelle observation  $i$  et les tats  $j$  chaque filtre de Kalman dj cr. La mise jour de chaque filtre s'effectue de la manire suivante : Pour chaque nouvelle observation  $i$ , on recherche le filtre dont l'tat Ensuite, les nouvelles observations sont utilises pour mettre jour les filtres de Kalman dont l'estimation est la plus proche selon cette matrice. Avant toute mise jour, on vifie que la distance entre l'observation et l'estimation du filtre ne dpasse pas un certain seuil. Pour chaque observation qui n'a pas pu tre associe un filtre dj existant, on cre alors un nouveau filtre.

### G. Reconnaissance Multi-vue

1) *Chanes de Markov Caches:* Le dplacement physique du robot produit une squence d'observations d'un mme objet, sous diffrents points de vues. On exploite l'information odomtrique entre les diffrentes vues pour renforcer l'estimation de la vue d'un objet. De cette manire, l'volution de la reconnaissance au cours du temps est reprsente par un processus stochastique, dont une modlisation possible consiste le traiter de faon discrte dans un espace d'tat. Ayant l'apriori que la dernire image et le dernier dplacement suffisent pour faire cette prdition (c'est--dire en respectant la prprit de Markov de premier ordre), le processus stochastique est modlise par une chane de Markov cache.

Concrtement, les tats cachs correspondent des vues d'objets prsents dans la base de donnes et dj stocks dans la

mmoire du robot. Cela constraint le nombre d'tats et garantie que la chane soit finie. Puis, une matrice de transition  $a_{i,j}$  dcrit l'volution du processus. C'est cette matrice de transition qui permet de prendre en compte l'odomtrie et la transition entre les vues d'un mme objets. Enfin, une autre matrice,  $P(y_1 | k)$ , dite matrice d'mission, estime la vraisemblance entre l'observation et les tats de la chane.

Plus prcisement,  $a_{i,j}$  est dfinie en fonction de l'angle  $\delta_{angle}$ , calcul par 7 qu'a parcouru le robot par rapport l'objet entre deux vues successives. Dans notre modle, on considre que  $a_{i,j}$  est nulle si  $i$  et  $j$  sont deux vues d'objets diffrents. Pour deux vues  $i$  et  $j$  d'un mme objet, spares d'une distance  $d$ , le poids accord  $a_{i,j}$  sera d'autant plus fort que  $\delta_{angle}$  et  $d$  sont proches. D'autre part, la matrice d'mission  $P(y_1 | k)$  correspond la similarit entre l'histogramme d'un objet segment  $y$  et celui d'une vue d'objet dans la base de donnees  $k$ , normaliss par l'quation 8. La similarit est calcule comme l'inverse de la distance Chi square dfinie la section II-E.

$$\begin{aligned}\vec{d}_0 &= p_0 - p_{obj} \\ \vec{d}_1 &= p_1 - p_{obj} \\ \delta_{angle} &= atan(\vec{d}_1) - atan(\vec{d}_0)\end{aligned}\quad (7)$$

La transformation des distances des histogrammes en probabilit est faite d'apr s la normalisation suivant :

$$P(y|x, database) = \frac{\sum_a d_a^x - d_y^x}{\sum_b \sum_c d_c^x - d_b^x} \quad (8)$$

o  $x$  est l'image de test et  $y$  un lment de la base de donnees. Dans le cas du plus proches voisin, la normalisation ne prend en compte que les  $k$  plus proches histogrammes, par opposition une approche *brute force*.

Une autre modlisation possible aurait t d'avoir une chane de Markov cache distincte pour chaque objet et ensuite dcider chaque pas de temps le processus le plus vraisemblable. Cette modlisation peut tre vue comme un sous-ensemble du cas prcdent o les transitions entre deux objets ne sont pas considres. Pourtant, il peut arriver que deux objets soit considrs comme positionns au mme endroit, ou bien que des objets mobiles fusionnent (par exemple, une personne qui viendrait s'asseoir sur une chaise, ou encore une personne qui commence marcher) <sup>1</sup>.

2) *Algorithme de Viterbi:* Reste donc extraire des informations de la modlisation Markovienne propos. La squence d'tats la plus vraisemblable qui pourrait avoir gnr les observations  $y_1, \dots, y_T$ , correspond normalement la squence d'objets reconnus. Afin de retrouver cette squence, aussi appelle chemin, on fait appel la programmation dynamique, et plus spcifiquement l'algorithme de Viterbi, d'o le nom chemin de Viterbi. L'algorithme retrouve de faon rcursive l'tat courant le plus probable, en prenant en compte seulement les observations jusqu' un instant donn et son estimation aux instants antrieurs. Ceci se traduit par les quations 9

<sup>1</sup>Le fait de se mettre en mouvement autre les formes d'une personne, ce qui rend possible sa dtction comme un nouvel objet.

$$\begin{aligned}V_{1,k} &= P(y_1 | k) \cdot \pi_k \\ V_{t,k} &= \max_{x \in S} (P(y_t | k) \cdot a_{x,k} \cdot V_{t-1,x})\end{aligned}\quad (9)$$

Ici,  $V_{t,k}$  reprsente la probabilit que la squence d'tats la plus probable finisse dans l'tat  $k$ , ayant gnr les observation l'instant  $t$ , tandis que  $\pi_i$  reprsente la probabilit initiale de se retrouver en chaque tat. Pour retrouver le chemin de Viterbi, il suffit de trouver le maximum de  $V_{t,k}$  :

$$x_T = \arg \max_{x \in S} (V_{T,x}) \quad (10)$$

### III. EXPERIMENTAL RESULTS

The proposed recognition system was deployed in a differential mobile robot, Wifibot V2, embedded with a RGB-D camera, Asus Xtion Pro Live. The algorithm architecture were implemented over ROS using PCL and OpenNi2 libraries. In the interest of validating the approach, the robot was initially taught objects aspects graphs and two sets of experiments were proposed to analyze the efficiency of the algorithm in real scenarios.

#### A. Object Database

First, twenty objects varying in size and form were selected to compose the robot knowledge database. The objects aspect graphs were composed by VFH features from eight equally distant viewpoints acquired from positioning the robot around the to be learn object 1.5 meters away. Each of the feature was labeled

#### B. Performance testing

The first experiment consist in a performance comparison between the single and multi image recognition techniques. In other words, this comparison attest whether the architecture is interesting or not, since having same performance by the cost of adding a complex post-processing and tracking modules is not interesting. Tours around each of the known objects arranged in 4 different poses.

#### C. Multiple object recognition

A second

To evaluate the recognition capability of our system, we use the Wifibot V2, equiped with an embedded computer. The RGB-D image aquisition is obtained with an Asus Xtion Pro Live sensor. We use ROS environment to acquire and process the data both from the camera and the odometry.

For our evaluation, twenty objects of various size and shapes (monitor, boxes, chairs, trashcans, persons, ...) were used to create the database. For each of them we moved the robot to make a complete circle around them separately. Eights view were obtained from each circle, so that the angular distance between each of them was 45 degrees. The point cloud and associated features were extracted and recorded for each view.

*1) Odometry Contribution:* In a first evaluation, another complete circle was made around objects to evaluate the recognition capacity of the system. For each objects, we made four successive circles starting at different positions around the object. The robot was recording for each circle between 20 and 30 frames at different viewpoints of the object.

La difficult de l'valuation vient, premirement, du fait que la base de donne a t ralise avec trs peu de vues, ce qui donne lieu de mauvaises reconnaissance mono-vue lorsque le point de vue observ se situe entre deux vues de la base de donne. De plus, la vitesse de dplacement peut gnrer des images plus floues lors des acquisitions et la modification de l'angle de la camra<sup>2</sup> apportent un obstacle en plus pour le *matching* de descripteurs dans le classificateur K-NN.

Un exprience typique est illustre dans l'image 5 :

La premire ligne correspond la squence d'images vues par le robot chaque instant de temps, et donc, l'objet tre reconnu. La seconde ligne, donne par l'algorithme de reconnaissance, quivaut la vue la plus probable de l'objet reconnue par le K-plus proches voisins. Il est intressant remarquer que l'invariance rotation du descripteur trompe l'estimation de l'orientation en prenant son correspond nantiomorphe dans le premier carr rouge. Autrement, le dos du pinguin tant une grande surface presque plane, il est partiellement retir par l'tape de segmentation. Ainsi, le nuage de points resultant de ce point de vue n'est pas suffisamment complet pour caractriser correctement l'objet, ce qui induit une mauvaise reconnaissance dans le carr bleu. Au final, on remarque que le traitement apport par la chane de Markov cache permet de corriger les problmes d'une base de donne relativement sparse avec des possibles erreurs de segmentation, permettant la correction simultane de la reconnaissance de l'objet et de son orientation en ligne.

Le rsultat de l'valuation peut tre reprsent sous forme d'une courbe : le nombre d'observations pour un mme objet en abscisse, par rapport aux pourcentage de reconnaissance d'objet et de vue, pour les reconnaissance mono et multi-vues.

La courbe de la figure 6 permet de conclure, tout d'abord, que l'algorithme de reconnaissance multi-vues est plus performant que sa correspondante mono-vue lorsque le nombre d'observations augmente suffisamment.<sup>3</sup>. Dans un deuxime temps, on constate que l'estimation de l'orientation de l'objet est plus difficile estimer que sa reconnaissance, que cela soit pour la reconnaissance mono-vue ou multi-vues.

#### D. Suivi et reconnaissance multi-cibles

La deuxime exprimentation consiste placer des objets prsents dans la base de donnees dans une pice et conduire le robot en faisant en sorte qu'il les regarde sous plusieurs

<sup>2</sup>L'angle entre la base et la tte de la camra Asus Xtion est facilement modifi.

<sup>3</sup>Le fait que la reconnaissance mono-vue chute avec le nombre de vues peut tre li au fait que pour les premires secondes de l'exprience, le robot est immobile et correctement plac. Lors du dplacement du robot, les prises de vues sont globalement moins bonnes et font chuter le taux de reconnaissance.

points de vues diffrents. Ce scenario est beaucoup plus complexe que celui d'avant. D'abord les objets s'occultent les uns les autres, donnant lieux de mauvaises segmentations. Ensuite, le suivi des objets est beaucoup plus complexe, car des objets proches peuvent tre confondus.

La carte finale donne par l'algorithme et localisant le robot ainsi que les objets dans le repre absolu est reprsente la figure 7. Une photo de la pice aves les objets disposs comme dans l'exprience est affiche en ??.

Les rsultats de l'exprience montrent que le systme fonctionne encore, mme dans des cas beaucoup plus complexes. On notera toutefois une chute des taux de reconnaissance. Quelques amliorations proposes dans la section suivante pourraient tre utiles pour rendre le systme plus robuste:

#### REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] Ros documentation. Tabletop object detector.
- [7] 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.
- [8] 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.



Fig. 5. **Experiment typique** - Reconnaissance multi-vue corrige des ambigus malgr la mauvaise segmentation lors de la cration de la base.

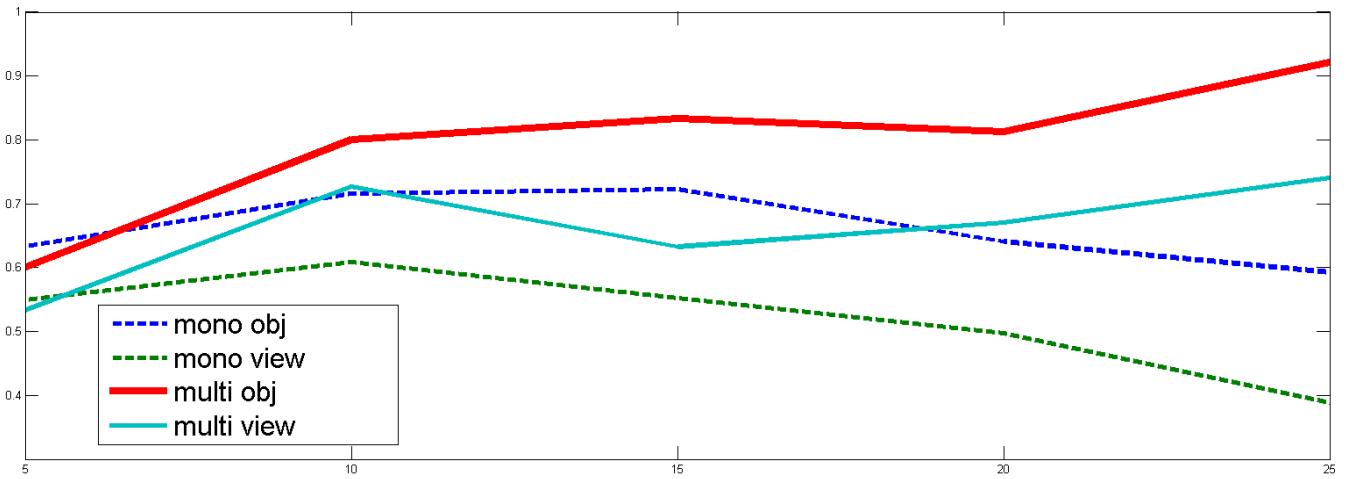


Fig. 6. **Rsultat de lvaluation** - Les courbes en pointill reprsentent la reconnaissance base sur une seule image (mono-vue), tandis que les courbes pleines correspondent au systme multi-vues. Avec un nombre rduit d'observations, la reconnaissance mono-vue tend tre lgrement plus performante. Lcart se creuse mesure que le nombre d'observations augmentent, jusqu' un cart de 33 % une fois le tour complet de l'objet effectu, soit 92% de russite pour l'estimation de l'objet et 74 % pour l'estimation de l'orientation.

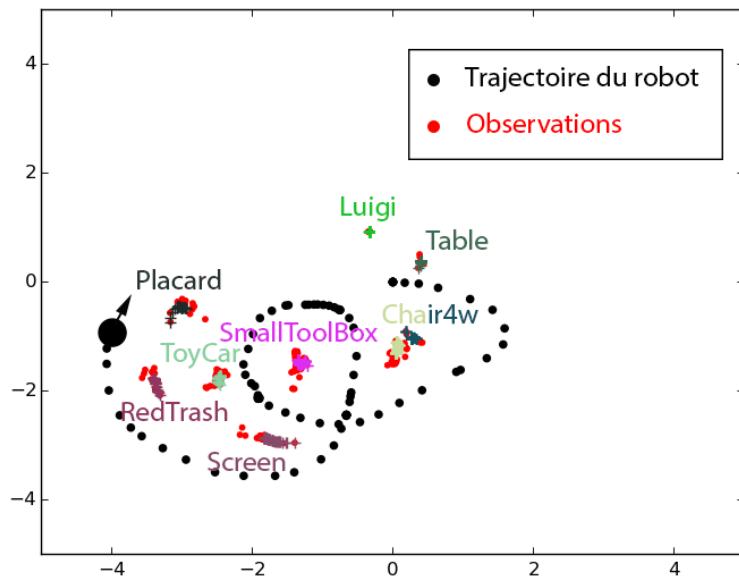


Fig. 7. **Rsultat du suivi multi-cibles** - Les points rouges correspondent aux centrodes des objets segments. Chaque croix de couleur reprsente une estimation de position d'un filtre de Kalman, avec une couleur par filtre. En thorie, un filtre devrait tre associ chaque objet. En noir la trajectoire du robot au cours du temps.



Fig. 8. **Reconnaissance Multi-cible** - Quatre des cinq objets présents dans la scène ont été correctement reconnus avec une estimation d'orientation raisonnable. Le premier objet (une personne) était trop près du ventilateur, les deux ont donc été confondus dans le multi-tracking.