Navigation, perception et apprentissage pour la robotique

David Filliat

Habilitation à Diriger des Recherches

Présentée à l'Université Pierre et Marie Curie
Spécialité « Informatique »

par
David Filliat

Navigation, perception et apprentissage pour la robotique

Soutenue le 12 juillet 2011

devant le jury composé de :

Pr.	Philippe Bidaud	(président du jury)
Dr.	Raja Chatila	(examinateur)
Pr.	Philippe Gaussier	(rapporteur)
Dr.	Simon Lacroix	(examinateur)
Pr.	Justus Piater	(rapporteur)
Pr.	Philippe Tarroux	(rapporteur)

RÉSUMÉ

Nous avons mené des travaux de recherche principalement dans les domaines de la navigation, la perception et l'apprentissage pour la robotique mobile. Ces travaux, orientés vers une robotique « cognitive », ont pour objectif général de permettre aux robots de s'adapter à leur environnement, en fournissant les primitives de base telles que l'espace libre, la position ou la présence d'objets nécessaires au choix des actions. Une grande partie de ces travaux sont guidés par une inspiration biologique essentiellement fonctionnelle, s'inspirant de capacités trouvées dans la nature, sans chercher à en reproduire précisément le fonctionnement.

La navigation, plus particulièrement la cartographie, a été jusqu'à présent le thème principal de nos travaux. Durant notre thèse, nous avons développé une méthode de cartographie utilisant un filtre bayésien, appliqué à une structure de carte et à des perceptions inspirées des connaissances biologiques sur les capacités de navigation du rat (Filliat and Meyer, 2000; Filliat, 2001; Filliat and Meyer, 2002). L'intérêt de cette approche est de permettre, à partir de capteurs très simples, une localisation globale durant la cartographie, apportant ainsi une bonne robustesse à la navigation, au prix d'une exploration relativement lente. Cette inspiration biologique s'est ensuite effacée dans les travaux menés à la Direction Générale pour l'Armement ou nous avons participé à la mise en place d'un démonstrateur utilisant des techniques de cartographie classiques à base de télémétrie laser et d'évitement d'obstacles en environnement dynamique (Dalgalarrondo et al., 2004). Depuis 2005 à l'ENSTA ParisTech, nos travaux se sont orientés sur les problèmes de navigation topologique, avec une approche de navigation topologique par apprentissage (Filliat, 2007; Filliat, 2008) dans laquelle un utilisateur désigne les pièces à reconnaître et montre le chemin entre les différentes pièces. Nous avons également développé une approche de cartographie topologique utilisant un algorithme de détection de fermetures de boucles qui permet de détecter le retour d'un robot à une position connue (Angeli et al., 2008a; Angeli et al., 2009; Bazeille and Filliat, 2011). Enfin, ces travaux se sont maintenant étendus, depuis 2009 dans le cadre du projet ANR PACOM (Filliat et al., 2009), à la problématique de la cartographie sémantique (Jebari et al., 2011). L'objectif est d'obtenir des modèles de l'environnement contenant des informations de plus haut niveau ; en particulier des informations plus proches de celles utilisées par l'humain, telles que les pièces ou les objets présents dans l'environnement.

Au niveau de la perception, certains de ces travaux ont fait appel à la télémétrie laser, qui est bien adaptée à la navigation. Ils sont néanmoins axés principalement sur l'utilisation de la vision. En particulier, nous nous sommes intéressés au problème de la représentation de l'information visuelle qui est essentielle pour apporter une robustesse au bruit tout en fournissant l'information nécessaire aux applications. Nous avons ainsi développé une approche incrémentale inspirée des modèles de « sac de mots » visuels que nous avons appliqué à la localisation qualitative (Filliat, 2007), topologique (Angeli et al., 2008b), au guidage visuel (Filliat, 2008) et à la reconnaissance d'objets (Rouanet et al., 2009). En collaboration avec Pierre-Yves Oudeyer nous avons étendu cette représentation à la reconnaissance auditive (Mangin et al., 2010) et audio-visuelle (ten Bosch et al., 2011). Nous nous sommes également inté-

ressés au problème de la perception active afin d'améliorer les capacités de localisation (Filliat and Meyer, 2000; Filliat, 2007) et d'améliorer la robustesse du guidage visuel (Filliat, 2008).

Enfin, la plupart de ces travaux ont fait appel à des méthodes d'apprentissage pour apporter de l'adaptabilité à la localisation, à la cartographie ou à la reconnaissance d'objets. Nous avons principalement travaillé avec des méthodes Bayésiennes et nous avons notamment développé des méthodes actives, permettant au robot de sélectionner les exemples d'apprentissage pour améliorer ses performances. Ces méthodes permettent également de profiter d'interactions avec l'utilisateur pour adapter les concepts appris par le robot. Nous les avons appliquées à la reconnaissance de pièces (Filliat, 2007; Filliat, 2008) et d'objets (Rouanet et al., 2009). Nous avons également appliqué la technique de factorisation en matrices non-négatives, une méthode d'apprentissage non supervisée, pour la reconnaissance audio-visuelle d'objets (ten Bosch et al., 2011). Cette dernière application se place dans le cadre de la robotique développementale où nous cherchons à nous inspirer de l'homme pour créer des méthodes d'apprentissage intuitives et à long terme pour la robotique, approche que nous développons actuellement dans le cadre du projet ANR MACSi (Sigaud et al., 2010).

Dans la continuité de ces travaux, nous souhaitons poursuivre nos recherches sur le thème de la navigation sémantique et de l'apprentissage pour la perception dans le cadre de la robotique développementale. Ces recherches auront pour objectif commun de fournir au robot ou à son utilisateur des modèles d'environnement riches et contenant des informations utiles à l'analyse de la situation ou aux tâches du robot. Ces méthodes s'appliqueront essentiellement dans le cadre de la navigation en milieu intérieur ou urbain et à la robotique de service ou d'assistance, en interaction directe avec l'homme.

ABSTRACT

We conducted research mainly in the areas of navigation, perception and learning for mobile robots. These studies, oriented toward a cognitive approach to robotics have the overall goal of allowing robots to adapt to their environment, providing basic primitives such as open space, position, or the presence of objects necessary to choose actions. A large part of this work is inspired by capabilities found in nature, but without trying to reproduce exactly the biological systems inner functioning.

Navigation, especially mapping, has so far been the focus of our work. During our PhD thesis, we developed a mapping method based on a Bayesian filter, applied to map structure and perceptions inspired from biological knowledge about the navigation capabilities of the rat (Filliat and Meyer, 2000; Filliat, 2001; Filliat and Meyer, 2002). The advantage of this approach is to allow, from simple sensors, global localization during mapping, providing a good robustness to kidnapping, at the price of a relatively slow exploration. This biological inspiration then disappeared during our work at the Direction Générale pour l'Armement where we participated in the development of a demonstrator using conventional mapping techniques based on laser ranging and obstacle avoidance in dynamic environments (Dalgalarrondo et al., 2004). Since 2005, at ENSTA ParisTech, our work has focused on the problem of topological navigation. We developed a learning-based topological approach (Filliat, 2007; Filliat, 2008) in which a user shows the rooms to be recognized and shows the path between the different rooms. We have also developed a topological mapping approach using a loop closure detection algorithm that can detect the return of a robot to previously visited locations (Angeli et al., 2008a; Angeli et al., 2009; Bazeille and Filliat, 2011). Finally, this work has now expanded since 2009 as part of the PACOM (Filliat et al., 2009) project to the problem of semantic mapping (Jebari et al., 2011). The objective is to obtain models of the environment that include higher-level information, in particular information closer to those used by humans, such as rooms or objects found in the environment.

In terms of perception, some of these works have used laser ranging, which is well suited to navigation. However, they are focused primarily on the use of vision. In particular, we investigated the problem of representation of visual information that is essential to provide robustness to noise while providing necessary information to applications. We have developed an incremental approach inspired by the "bag of visual words" model, which we applied to qualitative localization (Filliat, 2007), topological localization (Angeli et al., 2008b), visual guidance (Filliat, 2008) and object recognition (Rouanet et al., 2009). In collaboration with Pierre-Yves Oudeyer we have extended this representation to the auditory (Mangin et al., 2010) and audio-visual (ten Bosch et al., 2011) recognition. We also studied the problem of active perception in order to improve localization capabilities (Filliat and Meyer, 2000; Filliat, 2007) and improve the robustness of visual guidance (Filliat, 2008).

Finally, most of these studies have used learning methods to provide adaptability to localization, mapping or object recognition. We mainly worked with Bayesian models and we have developed active methods, allowing a robot to select the training samples to improve its performance. These methods can also benefit from interac-

tions with the user to adapt the concepts learned by the robot. We applied them to the recognition of rooms (Filliat, 2007; Filliat, 2008) and objects (Rouanet et al., 2009). We also applied non-negative matrix factorization, an unsupervised learning method, for audio-visual recognition of objects (ten Bosch et al., 2011). This last application was developed in the context of developmental robotics, where we take inspiration from humans to create intuitive and long term learning methods, an approach we are currently developing within the MACSi project (Sigaud et al., 2010).

In line with this work, we plan to continue our research on the topic of semantic navigation and learning for perception in the context of developmental robotics. This research will seek to provide rich environmental models to the robot or its user. These models will contain useful information for analysing the situation and for the robot tasks. These methods will be mainly applied in the context of navigation in indoor or urban environments and to service or assistive robotics, in direct interaction with humans.

TABLE DES MATIÈRES

1 INTRODUCTION 1
1.1 Contexte scientifique . . . 1
1.2 Thèmes de recherche . . . 2
1.3 Organisation du manuscrit . . . 3
2 NAVIGATION 5
2.1 Navigation topologique . . . 5
2.1.1 Filtrage Bayésien . . . 6
2.1.2 Navigation bio-inspirée . . . 6
2.1.3 Localisation et guidage qualitatifs . . . 10
2.1.4 Détection de fermeture de boucle . . . 13
2.1.5 Cartographie topo-métrique . . . 15
2.2 Navigation métrique et sémantique . . . 17
2.2.1 Démonstrateur de robot de reconnaissance . . . 17
2.2.2 Cartographie sémantique . . . 18
3 PERCEPTION 21
3.1 Perception pour la robotique . . . 22
3.1.1 Problème de la représentation . . . 22
3.1.2 Caractérisations bas-niveau . . . 22
3.1.3 Sacs de mots visuels incrémentaux . . . 23
3.1.4 Représentations intermédiaires . . . 25
3.1.5 Perception active . . . 26
3.2 Applications développées . . . 27
3.2.1 Reconnaissance d'objets . . . 27
3.2.2 Reconnaissance audio-visuelle . . . 28
3.2.3 Segmentation sémantique . . . 30
4 APPRENTISSAGE 33
4.1 Techniques mises en oeuvre . . . 34
4.1.1 Méthodes Bayésiennes . . . 34
4.1.2 Apprentissage non supervisé . . . 34
4.1.3 Apprentissage actif et interactif . . . 35
4.2 Approche développementale de la robotique . . . 37
4.2.1 Inspiration biologique . . . 37
4.2.2 Structuration de l'environnement . . . 38
5 PROJET DE RECHERCHE 41
5.1 Navigation sémantique . . . 41
5.1.1 Projet PACOM . . . 41
5.1.2 Cartographie topologique visuelle . . . 41
5.1.3 Navigation sémantique multi-modale . . . 42
5.2 Approche développementale de la perception . . . 42
5.2.1 Projet MACSi . . . 42

5.2.2 Apprentissage multi-modal 43
5.3 Domaines d'application 43

BIBLIOGRAPHIE 45

A CURRICULUM VITAE 53
B SÉLECTION D'ARTICLES 59
B.1 Global localization and topological map-learning for robot navigation . 61
B.2 Interactive learning of visual topological navigation 71
B.3 A Fast and Incremental Method for Loop-Closure Detection 79
B.4 Incremental topo-metric SLAM using vision and robot odometry 91
B.5 Controlling the autonomy of a reconnaissance robot 99
B.6 Multi-sensor semantic mapping and exploration of indoor environments 111
B.7 An integrated system for teaching new visually grounded words 117
B.8 A bag-of-features framework for incremental learning of speech 125

1 INTRODUCTION

Ce mémoire synthétise les travaux de recherche menés depuis notre thèse, réalisée au Laboratoire d'Informatique de Paris VI, dans l'équipe AnimatLab et soutenue en décembre 2001. Ils ont été réalisés dans un premier temps au Centre Technique d'Arcueil où nous avons été Expert en Robotique pour la Direction Générale pour l'Armement de 2002 à 2005, puis, pour la plus grande partie, à l'Ecole Nationale Supérieure de Techniques Avancée où nous sommes enseignant-chercheur depuis 2005 au sein de l'Unité Électronique et Informatique.

1.1 CONTEXTE SCIENTIFIQUE

La robotique est un domaine scientifique vaste et pluri-disciplinaire, partant des aspects mécaniques, passant par l'automatique et allant jusqu'aux aspects de plus haut-niveau tels que la perception, la modélisation de l'environnement et la décision. En plus des défis liés à tous ces domaines, l'intégration de tous ces éléments pour réaliser un robot est également un problème à part entière. Parmi ces thèmes multiples, nos travaux de recherche se focalisent essentiellement sur les aspects « cognitifs » de la robotique, reliés essentiellement à l'informatique, à la vision par ordinateur et à l'Intelligence Artificielle. Nous nous sommes également intéressés à la problématique de l'intégration au travers du développement de démonstrateurs pour l'exploration d'environnements.

Plus particulièrement, nous nous sommes intéressés à la perception qui est un élément essentiel pour permettre une interaction riche entre le robot et son environnement tant matériel que social. En effet, la perception est souvent traitée dans le cadre d'une tâche très précise telle que la navigation, la reconnaissance d'objets ou l'interaction avec l'homme. Toutefois, pour progresser réellement dans l'interaction, des capacités de perception beaucoup plus riches et génériques seront nécessaires. Ainsi pour la navigation par exemple, la plupart des représentations sont spécifiques et contiennent uniquement les obstacles à la navigation du robot ou les points de repère pour sa localisation. Ces représentations sont progressivement étendues par des éléments plus *sémantiques* tels que les types d'obstacles ou de pièces. Ces éléments permettent d'envisager des comportements plus complexes (par example pousser un obstacle reconnu comme mobile) ou des interactions plus naturelles avec l'humain en prenant en compte le contexte (par example en allant chercher une boisson directement dans la cuisine sans que l'utilisateur ait besoin de le préciser). Nos travaux ont donc porté sur des problèmes de perception spécifique et ont évolué au cours du temps vers des problèmes plus génériques dans ce cadre de perception sémantique.

Certains de nos travaux, en particulier notre premier modèle de navigation et nos travaux sur l'apprentissage et la perception, s'inspirent en partie de la biologie. D'une

manière générale, l'inspiration biologique n'est pas forcement une bonne approche pour créer des mécanismes artificiels (la roue ou l'avion sont de bons contre-exemples), mais nous pensons cependant qu'elle indique des pistes de recherche intéressante : l'homme n'aurait peut-être pas essayé de construire des avions si les oiseaux n'existaient pas. Cette inspiration biologique reste donc de très haut-niveau dans nos travaux, fournissant plus des problèmes intéressant à étudier que des solutions. Pour la robotique, la nature (l'homme en particulier) fournit un bon modèle de référence pour des fonctions complexes telles que la perception de l'environnement ou l'apprentissage. Il ne s'agit donc pas en général de comprendre ou reproduire le fonctionnement exact de la nature (comme par exemple en neurosciences computationelles), ni de chercher a reproduire précisément une fonction naturelle indépendamment de l'intérêt de cette fonction pour des applications réelles en robotique. Cette inspiration biologique reste donc un guide permettant d'orienter des recherches, sans en être une justification.

1.2 THÈMES DE RECHERCHE

La navigation pour la robotique mobile, et plus particulièrement la localisation et la cartographie, a été jusqu'à présent le thème principal de nos travaux. Nous avons développé différents modèles de navigation topologique utilisant la vision, soit inspirés de la biologie, soit inspiré de problématiques de traitement d'images. Nous avons également participé au développement d'un démonstrateur de robot de reconnaissance intégrant une méthode de cartographie métrique utilisant la télémétrie laser. Enfin, dans le cadre du projet ANR PACOM, nos travaux se sont étendus récemment à la problématique de la cartographie sémantique multi-modale dont l'objectif est d'obtenir des modèles de l'environnement contenant des informations de plus haut niveau ; en particulier des informations plus proches de celles utilisées par l'humain, telles que les pièces ou les objets présents dans l'environnement.

Au niveau de la perception, certains de ces travaux ont fait appel à la télémétrie laser, qui est bien adaptée à cette problématique. Ils sont néanmoins axés principalement sur l'utilisation de la vision. En particulier, nous nous sommes intéressés au problème de la représentation de l'information visuelle qui est essentiel pour apporter une robustesse au bruit tout en fournissant l'information nécessaire aux applications. Nous avons ainsi développé une approche incrémentale inspirée des modèles de « sac de mots visuels » que nous avons appliqué à la navigation, à la reconnaissance d'objets et étendu à la reconnaissance auditive. Nous nous sommes également intéressés au problème de la perception active afin d'améliorer les capacités de reconnaissance. Enfin, nous développons des méthodes de perception multi-modales permettant de fournir les informations de haut-niveau nécessaires à la cartographie sémantique.

La plupart de nos travaux ont fait appel à des méthodes d'apprentissage afin d'apporter de l'adaptabilité à la localisation, à la cartographie ou à la reconnaissance d'objets. Nous avons notamment développé des méthodes actives, permettant au robot de sélectionner les exemples d'apprentissage pour améliorer ses performances. Ces méthodes permettent également de profiter d'interactions avec l'utilisateur pour adapter

les concepts (pièces, objets) appris par le robot. Enfin, nous avons utilisé des méthodes d'apprentissage non supervisé, pour la reconnaissance audio-visuelle d'objets. Cette dernière méthode se place dans le cadre de la robotique développementale où nous cherchons à nous inspirer de l'homme pour créer des méthodes d'apprentissage intuitives et à long terme pour la robotique, approche que nous développons actuellement dans le cadre du projet ANR MACSi.

1.3 ORGANISATION DU MANUSCRIT

Ce manuscrit est organisé par thématiques selon les trois axes mentionnés précédemment : la navigation (chapitre 2), la perception (chapitre 3) et l'apprentissage (chapitre 4). Ainsi, une application particulière pourra être présentée dans ces différents chapitres selon chacun de ces différents aspects. Pour conclure, le chapitre 5 présente nos projets de recherche en cours et leur perspectives pour les prochaines années.

2 NAVIGATION

Les méthodes de cartographie et de localisation en robotique reposent sur plusieurs éléments : un choix de capteur, un type de carte et une méthode d'estimation. Concernant ce dernier point, nous avons presque toujours utilisé des filtres Bayésiens classiquement utilisés en robotique (Thrun et al., 2005). Nos recherches ont été concentrées sur les types de cartes utilisables comme support de ces filtres et sur les traitements associés des données capteurs en vision ou en télémétrie.

Historiquement, les systèmes de navigation peuvent être grossièrement classés en fonction des types de cartes utilisées (Filliat and Meyer, 2003; Meyer and Filliat, 2003). Les cartes *métriques* (e.g. (Moravec and Elfes, 1985; Chatila and Laumond, 1985)) permettent d'enregistrer dans un repère commun les obstacles ou les points de repère utilisés pour la navigation. Les cartes *topologiques* (e.g. (Kuipers and Byun, 1991)) enregistrent un ensemble de lieux accessibles au robot ainsi que le moyen de passer d'un lieu à ses voisins sous forme d'un graphe. Les cartes métriques ont l'avantage de fournir une estimation plus précise de la position, tandis que les cartes topologiques fournissent directement une segmentation de l'espace en différents lieux qui peut être intéressante pour la navigation ou l'interaction avec l'homme. De nombreuses approches hybrides ont également été développées (e.g. (Kuipers et al., 2004)) pour tirer partie des deux approches et plusieurs systèmes récents utilisant des ensembles de poses relatives rendent leurs frontières relativement floues (Agrawal and Konolige, 2008). Enfin, les méthodes de cartographie sémantiques intègrent hiérarchiquement ces types de cartes avec des informations supplémentaires telles que la nature des pièces ou les objets présents dans l'environnement (Galindo et al., 2005).

La cartographie et la localisation on été les thèmes les plus développés dans nos travaux. Nous nous sommes plus particulièrement intéressés à la navigation visuelle avec différentes approches topologiques et plus récemment à des approches de cartographie sémantique multi-modales.

2.1 NAVIGATION TOPOLOGIQUE

Nous avons développé plusieurs approches de navigation topologique utilisant des filtres Bayésiens pour estimer la position du robot au sein de la carte. Ces approches diffèrent essentiellement par la complexité du traitement des informations visuelles qui s'est progressivement accrue.

2.1.1 Filtrage Bayésien

Le filtrage bayésien est une méthode générique qui permet d'intégrer dans le temps des informations sur la position du robot venant des perceptions et des informations sur le mouvement venant, par exemple, de l'odométrie. C'est une méthode récursive, qui met à jour une distribution de probabilité de position du robot. Son équation générique en version discrète (Thrun et al., 2005) est :

$$p(X_t|Z^t,U^t) = \eta.\ \underbrace{p(Z_t|X_t)}_{\text{vraisemblance}}\ .\ \underbrace{\sum_{j=0}^{n} \underbrace{p(X_t|X_{t-1}=j,U_t)}_{\text{modèle de transition}}\ \underbrace{p(X_{t-1}=j|Z^{t-1},U^{t-1})}_{\text{probabilité } a\ priori}}_{\text{prédiction}}$$

où η est un facteur de normalisation, Z^t est l'ensemble des perceptions du robot depuis l'origine, Z_t est la perception à l'instant t, U^t l'ensemble des mesures d'odométrie (ou les contrôles envoyés au robot) depuis l'origine et U_t l'odométrie entre les temps $t-1$ et t.

Cette équation permet ainsi d'estimer la position du robot au travers de modèles probabilistes de l'odométrie $p(X_t|X_{t-1}=j,U_t)$ et des perceptions associées à la carte de l'environnement $p(Z_t|X_t)$. Nos différents travaux ont ainsi porté sur les représentations des états X_t et sur ces deux modèles probabilistes.

2.1.2 Navigation bio-inspirée

Travaux réalisés dans le cadre de notre thèse (Filliat, 2001; Filliat et al., 2004).

Pour ce modèle, nous nous sommes inspirés des connaissances sur les capacités de navigation des rats, notamment de l'existence dans l'hippocampe du rat de *cellules de lieux* (O'Keefe and Conway, 1978). Ces neurones ont la particularité de s'activer pour certaines positions du rat dans son environnement et fournissent donc une représentation interne de l'environnement, une *carte cognitive* (Tolman, 1948). De nombreuses études ont montré que l'activation de ces cellules dépend à la fois de la vision et de l'information de déplacement obtenue par le mouvement des pattes. Ceci rejoint le problème classique en robotique de fusionner des estimations de déplacement et des estimation de position venant de capteurs tels que les caméras ou les télémètres.

Il existe différents modèles robotiques biomimétiques de ces cellules de lieu (Arleo et al., 2004; Cuperlier et al., 2007). Pour notre part, nous nous sommes inspirés de ces données pour créer un modèle de navigation permettant de réaliser une localisation robuste et la cartographie d'environnements inconnus. Sans chercher une plausibilité biologique poussée, ceci nous a conduit à développer une carte topologique dense dont les noeuds sont irrégulièrement répartis dans l'environnement. Nous avons également utilisé une vision panoramique à très basse résolution, ainsi que la distance aux obstacles détectés sur notre robot par des sonars à courte portée (Figure 1). Le système complet est décrit dans (Filliat, 2001).

La localisation est assurée par un filtre très similaire à un filtre Bayésien classique. Un certains nombre de modifications ont cependant été ajoutées pour gérer les irrégularités de la carte. Globalement, ce filtre permet de fusionner les perceptions et les mesures d'odométrie pour fournir une estimation probabiliste robuste de la position.

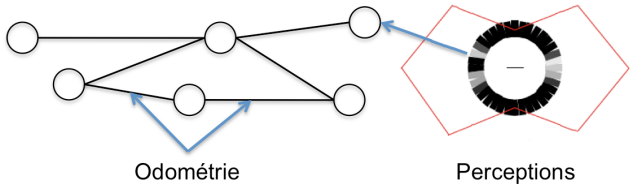

FIGURE 1.: Type de carte utilisé dans notre modèle de navigation bio-inspiré. Les nœud mémorisent les perceptions : une vue panoramique en niveaux de gris à très basse résolution (36 pixels de 10 degrès d'ouverture) et les données des sonars (ligne rouge). Les arêtes mémorisent les informations d'odométrie reliant deux positions.

Par similarité avec les cellules de lieu, nous avons appelé « activité » la probabilité de présence du robot en chaque noeud de la carte. L'équation de mise a jour est alors :

$$Act_i(t) = \eta P_{vis}(I_t|I_{N_i}) \sum_{j \in S_{N_i}} P_{od}(od|ji)\, Act_j(t-1)$$

où $P_{vis}(I_t|I_{N_i})$ est la vraisemblance des perceptions courantes à la position N_i, $P_{od}(od|ji)$ est la probabilité de l'odométrie od si le robot s'est déplacé du noeud j au noeud i et S_{N_i} est l'ensemble des noeuds connectés à N_i.

La vraisemblance des perceptions est estimée simplement à partir des différences entre image perçue et image mémorisée dans le noeud avec un modèle de bruit gaussien. La même équation est utilisée pour les données sonar :

$$P_{vis}(I_t|I_{N_i}) = \prod_{k=1}^{36} \exp\left(\frac{-\left(I_k^k - I_{N_i}^k\right)^2}{\sigma}\right)$$

L'estimation de la probabilité du déplacement od de distance r_{od} dans la direction θ_{od} si le robot s'était déplacé du noeud N_j au noeud N_i est également donné par un modèle de bruit gaussien :

$$P_{od}(od|ji) = exp\left(\frac{-(\theta_{od} - \theta_{ji})}{L^2}\right) \times exp\left(\frac{-(r_{od} - r_{ji})}{M^2}\right)$$

Cette méthode permet d'estimer de manière très fiable la position du robot avec des probabilités pour chaque noeud de la carte très similaire à l'activité des cellules de lieu des rats (Figure 2).

Pour pouvoir réaliser la cartographie d'un environnement inconnu, il faut de plus pouvoir déterminer si le robot se trouve à la position de la carte considérée comme la plus probable, ou s'il est sorti de la zone actuellement cartographiée. Ceci est difficile car ce dernier cas n'est pas pris en compte par le filtrage Bayésien. Nous avons utilisé une heuristique prenant en compte la somme des activités avant et après la prédiction de position par l'odométrie (Figure 3). Lorsque la diminution de cette somme est supérieure à un seuil, le robot est considéré comme étant en zone inconnue et un nouveau noeud est ajouté à la carte, initialisé avec les perceptions courantes du robot. Le nombre d'états du filtre Bayésien est donc augmenté également. Un algorithme de relaxation itératif (Duckett et al., 2000) est utilisé afin d'estimer la position absolue

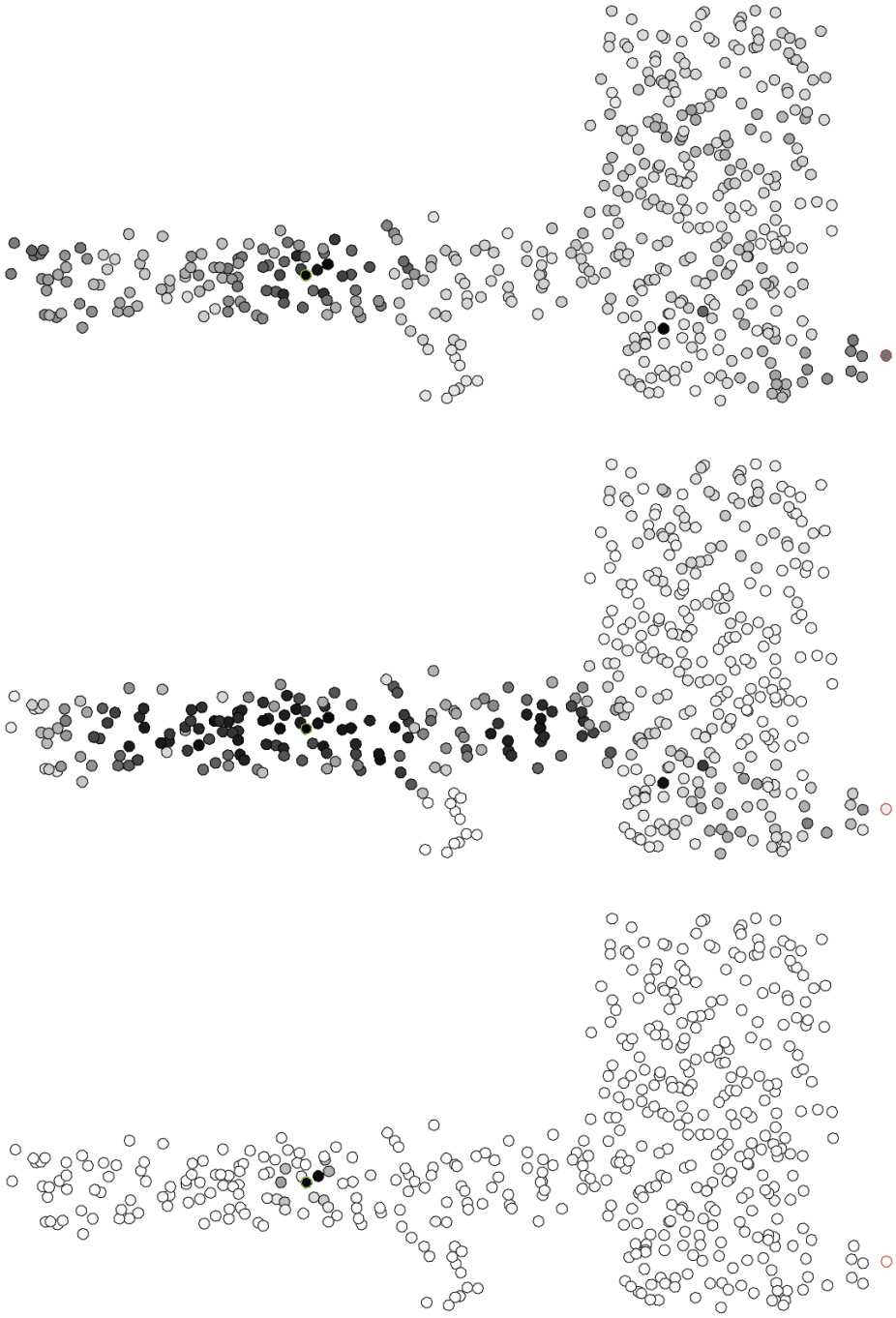

FIGURE 2.: Exemple d'activation estimées a l'aide de notre modèle. Haut : vraisemblance liée à la vision ; Milieu :vraisemblance liée aux sonars ; Bas : activité intégrée avec l'odométrie.

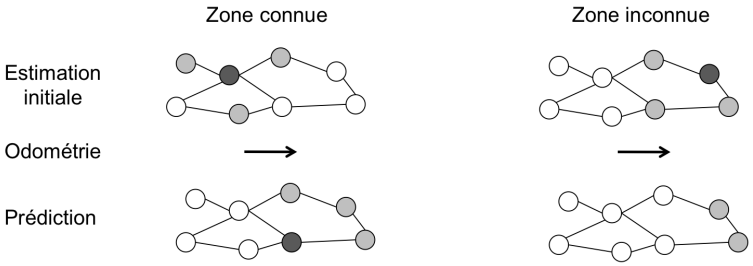

FIGURE 3.: Estimation de la nouveauté en fonction de la somme des activités des nœuds de la carte avant et après l'intégration de l'odométrie. En zone couverte par la carte, cette somme reste approximativement constante (Partie gauche), pour un lieu nouveau, cette somme diminue (Partie droite).

de chaque noeud qui est la plus cohérente avec l'ensemble des mesures d'odométrie intégrées dans la carte (Figure 4).

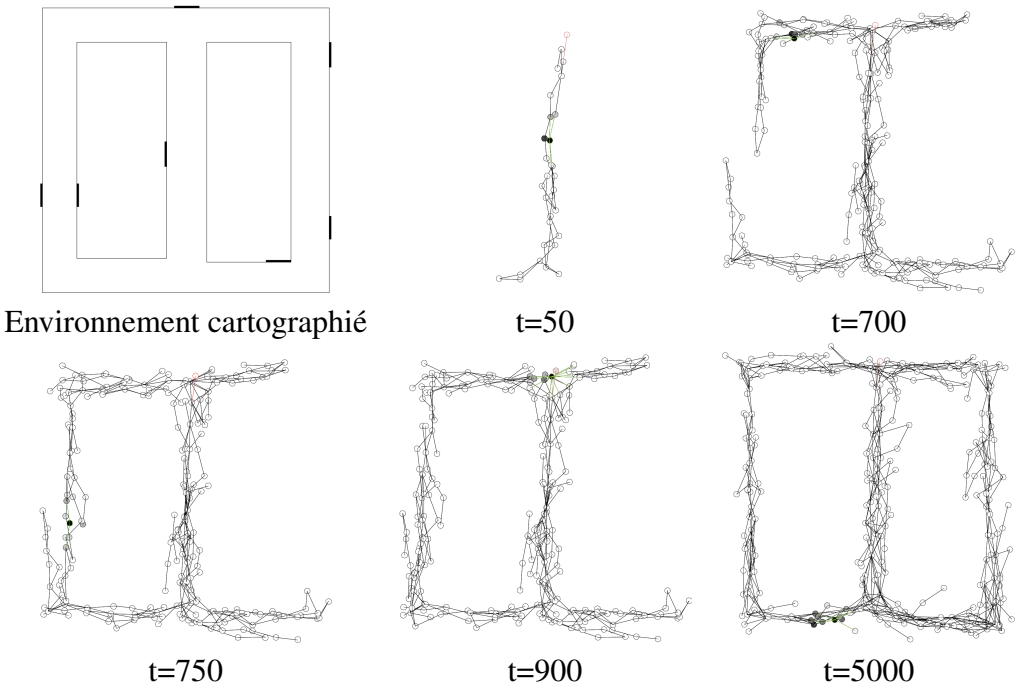

FIGURE 4.: Évolution de la carte lors de la cartographie d'un environnement cyclique. L'algorithme de relaxation permet de corriger les erreurs d'odométrie initialement introduites dans la carte.

Le modèle utilise une méthode de planification permettant de prendre en compte des obstacles dynamiques tels que des portes qui peuvent ponctuellement bloquer un passage et obliger le robot à prendre un autre chemin. Il intègre également une méthode de perception active dont nous parlerons au chapitre 3 qui permet d'étendre le fonctionnement du modèle à l'utilisation de caméra perspectives standards. Enfin, ce modèle a été validé en simulation et sur un robot réel.

Ce modèle permet donc une cartographie et une localisation robuste avec des capteurs très simples. Notons cependant qu'il requiert une exploration très lente et progressive de l'environnement pour limiter l'impact des erreurs d'odométrie et qu'il nécessite d'avoir une estimation séparée de la direction. Son architecture sera cependant réutilisée dans d'autres modèles que nous développerons par la suite (voir section 2.1.5) pour compenser ces limitations.

2.1.3 Localisation et guidage qualitatifs

Travaux réalisés dans le cadre des stages de Jose-Luis Susa et Florian Vichot (Filliat, 2007; Filliat, 2008).

Nous avons développé un second modèle de navigation topologique utilisant des méthodes de perception beaucoup plus efficaces et reposant sur l'interaction avec l'utilisateur pour la cartographie de l'environnement. Contrairement au modèle précédent, les noeuds de la carte topologique représentent des zones beaucoup plus vastes, par exemple les différentes pièces de l'environnement. Pour le passage d'une pièce à une autre, nous avons développé une méthode de guidage visuel apprise elle aussi en interaction avec l'utilisateur. Ce modèle est fonctionellement très proche des travaux de Giovannangeli et Gaussier (Giovannangeli and Gaussier, 2008) qui utilisent une architecture bio-inspirée.

Ce modèle repose sur la vision au travers d'une méthode de « sac de mots visuels ». Cette méthode permet de représenter des images par des histogrammes d'occurrence de caractéristiques locales quantifiées. Elle sera décrite plus en détails dans la section 3.1.3.

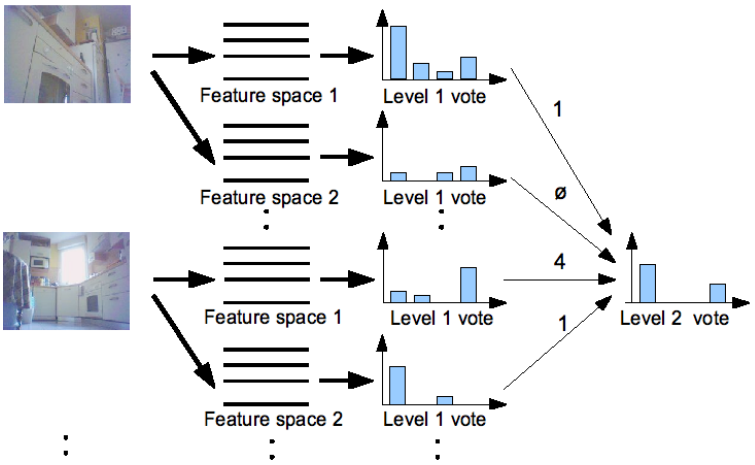

FIGURE 5.: Illustration de la méthode de vote utilisée pour prédire la pièce dans laquelle se trouve le robot.

A partir de la représentation d'une image par son sac de mots correspondant, une méthode de classification par vote permet de reconnaitre la pièce courante. Deux niveaux de vote sont utilisés afin d'intégrer plusieurs caractéristiques et plusieurs images (Figure 5) tout en filtrant les images non informatives. Pour le premier niveau, chaque mot détecté dans l'image courante vote pour les pièces dans lesquelles il a déjà été vu avec sa fréquence de document inversée (Sivic and Zisserman, 2003).

Ce coefficient a pour but de donner plus de poids aux mots visuels vus dans moins de pièces et donc plus discriminants :

$$idf = \log(N/n_i) / \log(N)$$

où N est le nombre de pièces total et n_i le nombre de pièces dans lesquelles le mot i a été vu. Cette méthode est similaire à une méthode Bayésienne naïve (section 4.1.1) si l'on considère le terme idf comme le logarithme de la probabilité d'apparition du mot dans la pièce.

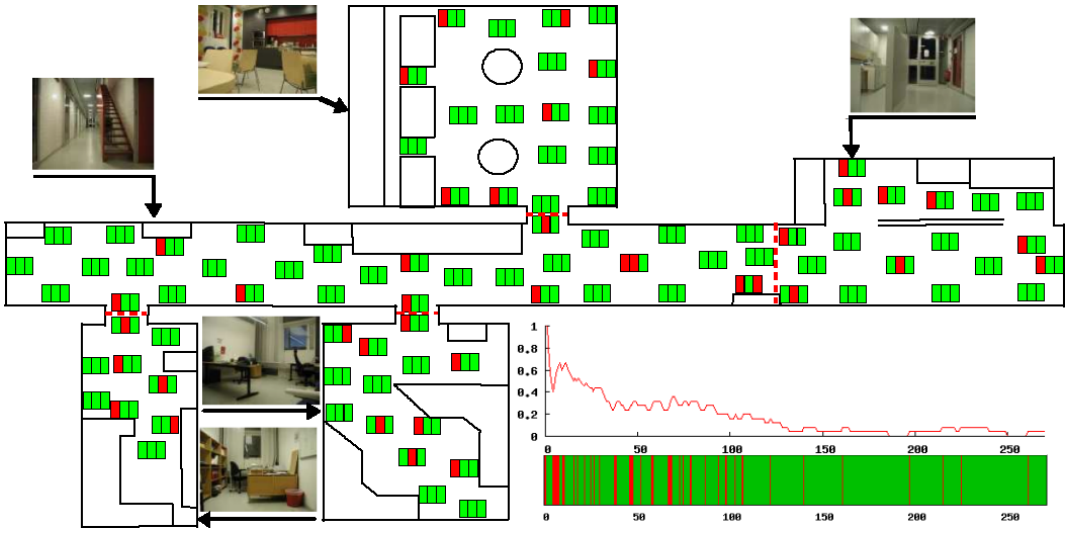

FIGURE 6.: Exemple de résultats de localisation et de cartographie sur la base INDECS (Pronobis et al., 2006). Pour chaque position, 3 rectangles montrent les 3 expériences de localisation successives effectuées depuis ce lieu avec différentes conditions d'éclairage. Un rectangle vert correspond à une localisation réussie, un rouge correspond à une erreur. Dans ce dernier cas, les images utilisées sont apprises. Les diagrammes en bas à droite montrent la succession temporelle des résultats de localisations correctes et incorrectes (en bas) et l'évolution du taux d'erreur sur les 25 dernières localisations (en haut).

La qualité du résultat du vote est estimée par l'écart entre la pièce ayant le meilleur score et la pièce ayant le second meilleur score, divisé par la somme des scores :

$$qualite = \frac{v_{Gagnant} - v_{Second}}{\sum_j v_j}$$

Si cette qualité est au dessus d'un seuil, l'image courante apporte une information et un vote pour la pièce gagnante est réalisé au deuxième niveau. Cette procédure est répétée avec de nouvelles caractéristiques d'images ou avec de nouvelles images jusqu'à ce que la qualité du vote de deuxième niveau dépasse un seuil ou qu'un nombre maximal d'images ait été traité. Si le seuil de qualité est dépassé, la pièce est reconnue, sinon le système demande à l'utilisateur quelle est la pièce courante afin de mettre a jour le modèle de l'environnement. Ce processus permet de converger vers un modèle correct de l'environnement qui permet de prédire correctement la position avec une précision supérieure à 90% après 180 tentatives de localisation (Figure 6).

Une méthode très similaire est utilisée pour apprendre une stratégie de guidage visuel qui permet au robot d'atteindre les portes dans chaque pièce. Par rapport à la prédiction des pièces, qui est un processus de catégorisation, il s'agit ici d'un problème de régression. En effet, depuis sa position courante, le robot doit prédire la direction à prendre pour atteindre le but. Ceci est réalisé avec une méthode de vote portant sur un ensemble de directions discrétisées. Pour l'apprentissage, l'utilisateur montre la bonne direction et tous les mots visuels détectés lui sont associés(Figure 7, en bas à gauche). Pour la prédiction, les mots visuels de l'image courante votent pour la direction associée lors de l'apprentissage et le robot ce déplace dans la direction majoritaire. Cette procédure permet d'apprendre une stratégie de guidage visuel robuste et efficace sur tout l'environnement (Figure 7, à droite). L'utilisation d'une méthode de perception active (voir section 3.1.5) et l'approche par sacs de mots visuels permet notamment une bonne robustesse face aux variations locales de l'environnement comme les personnes se déplaçant.

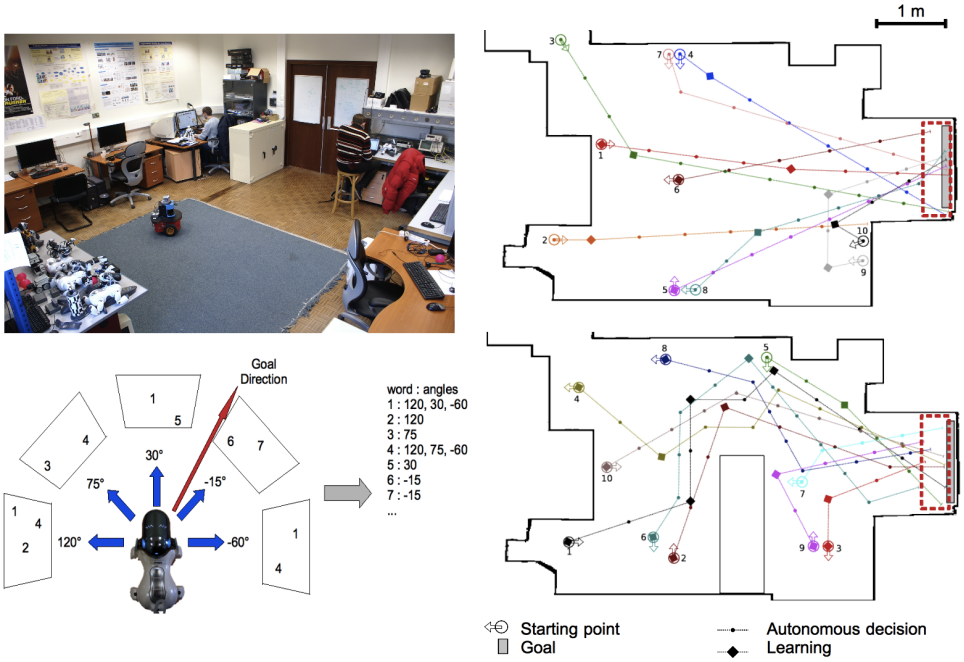

FIGURE 7.: Illustration de la méthode d'apprentissage de la direction (en bas à gauche). Résultat du guidage visuel dans notre laboratoire avec et sans obstacle central (à droite). Le but est le rectangle sur la droite de l'environnement.

Ce modèle permet donc une cartographie et un guidage robuste en utilisant uniquement la vision, en profitant d'une interaction forte avec l'utilisateur pour l'apprentissage. Les bonnes performances obtenues découlent directement de la robustesse de la représentation des images par la méthode de sacs de mots visuels et des méthodes de perception actives qui permettent de rechercher l'information pertinente. La localisation est cependant peu précise car il n'y a pas d'information sur la position au sein d'une pièce.

2.1.4 Détection de fermeture de boucle

Travaux réalisés dans le cadre des thèses d'Adrien Angeli et d'Alexandre Chapoulie en coopération avec Patrick Rives (Angeli et al., 2008a; Angeli et al., 2008b).

Nous avons développé une autre méthode de cartographie topologique utilisant uniquement la vision partant d'une approche très différente. Le problème qui nous a intéressé à l'origine était celui de la détection de fermeture de boucles, c'est à dire la détection du retour de la caméra près d'une position déjà visitée dans le passé. Ce problème est intéressant car il peut permettre de recaler un algorithme de cartographie métrique (e.g., (Lemaire et al., 2007)) dont l'estimation de la position aurait trop dérivé. Il peut également permettre d'initialiser l'estimation de position lorsque l'on met en fonctionnement le robot (Angeli et al., 2009).

Nous avons réalisé la détection de fermeture de boucle à l'aide d'un filtre Bayesien qui permet d'estimer la probabilité de l'évènement $S_t = i$ que l'image courante I_t viennent du même lieu que l'image passée I_i. Cette probabilité est estimée grâce à l'équation :

$$p(S_t|I^t) = \eta p(I_t|S_t) \sum_{j=0}^{t-p} p(S_t|S_{t-1} = j) p(S_{t-1} = j) \tag{2.1}$$

où $p(I_t|S_t)$ est la vraisemblance de l'image I_t pour le lieu S_t et $p(S_t|S_{t-1})$ le modèle d'évolution qui permet d'imposer une cohérence temporelle des détections.

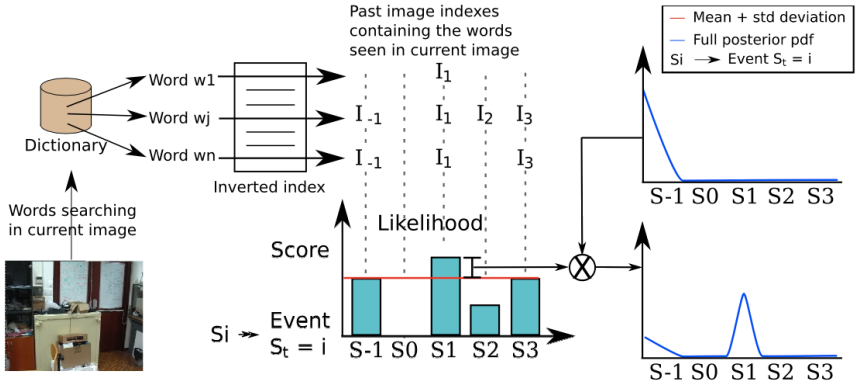

FIGURE 8.: Illustration de la méthode de calcul de la vraisemblance utilisant l'approche sac de mots visuels et l'index inversé.

Estimer la vraisemblance de l'image courante demande de comparer cette image avec toutes les images passées. Une implémentation naïve demanderait un temps linéaire en fonction du nombre d'images vues et serait rapidement inapplicable en temps réel. Nous avons donc utilisé notre approche de sacs de mots visuels incrémentaux (voir section 3.1.3) qui permet au travers d'un index inversé d'obtenir très rapidement un score de similarité entre l'image courante et les images passées (Figure 8). La fonction de vraisemblance est ensuite construite en seuillant et normalisant ce score de vraisemblance. Le temps de calcul est ici approximativement proportionnel au nombre de mots visuels de l'image courante. Pour l'estimation de l'évolution, en l'absence d'information sur le mouvement de la caméra, une simple diffusion gaussienne vers les états voisins est utilisée.

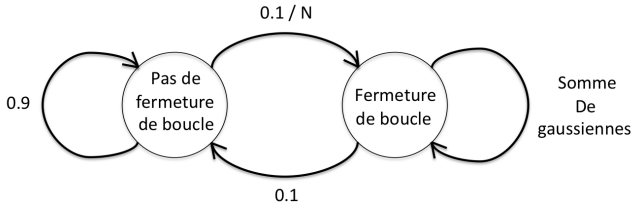

FIGURE 9.: Modèle d'évolution prenant en compte l'évènement « pas de fermeture de boucle ».

Comme pour notre modèle biologiquement inspiré (section 2.1.2), il est nécessaire à chaque étape de décider si la position visitée est connue ou non. Dans ce modèle, ceci est fait différemment en créant un état supplémentaire correspondant à l'évènement « la caméra est dans une position inconnue ». Cet état est associé à une image virtuelle correspondant à l'image moyenne perçue dans l'environnement. Cette image est construite à partir des statistiques sur les mots visuels. Sa vraisemblance est simplement estimée comme pour toutes les autres images. Un modèle d'évolution particulier permet d'intégrer cet évènement avec les autres noeuds de la carte (Figure 9).

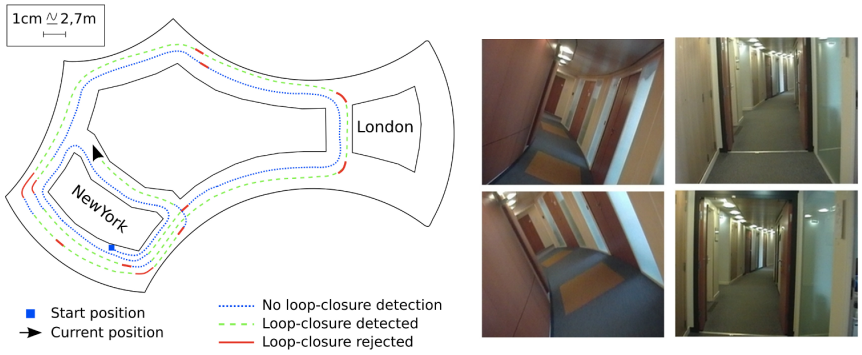

FIGURE 10.: Exemple de résultats de la méthode de détection de fermeture de boucles dans un environnement intérieur. Le schéma de gauche illustre la trajectoire et les résultats de détection, les images de droite illustrent deux paires d'images reconnues comme venant de la même position.

A partir des hypothèses de fermeture de boucle dont la probabilité dépasse un certain seuil, une vérification de la géométrique épipolaire de la correspondance des images est réalisée pour limiter le taux de fausses alarmes. Cette étape s'avère nécessaire car la méthode de sacs de mots visuel est très rapide, mais ignore la position des points d'intérêt dans l'image et peut conduire à des erreurs dans des environnements très ambigus. Ce modèle permet d'obtenir de très bonnes performances en environnement intérieur (Figure 10) comme en environnement extérieur (Angeli et al., 2008a).

Dans le cadre de la thèse d'Alexandre Chapoulie, co-dirigée avec Patrick Rives de l'équipe INRIA AROBAS, nous étendons actuellement ce modèle à l'utilisation de caméras panoramiques au travers de l'utilisation d'une représentation sphérique des images. Cette représentation permet d'intégrer efficacement des informations de contexte autour des mots visuels afin d'améliorer le taux de détection, notamment lors

du retour dans une zone connue avec une direction différente. Elle permet également de se passer d'une étape finale de validation géométrique.

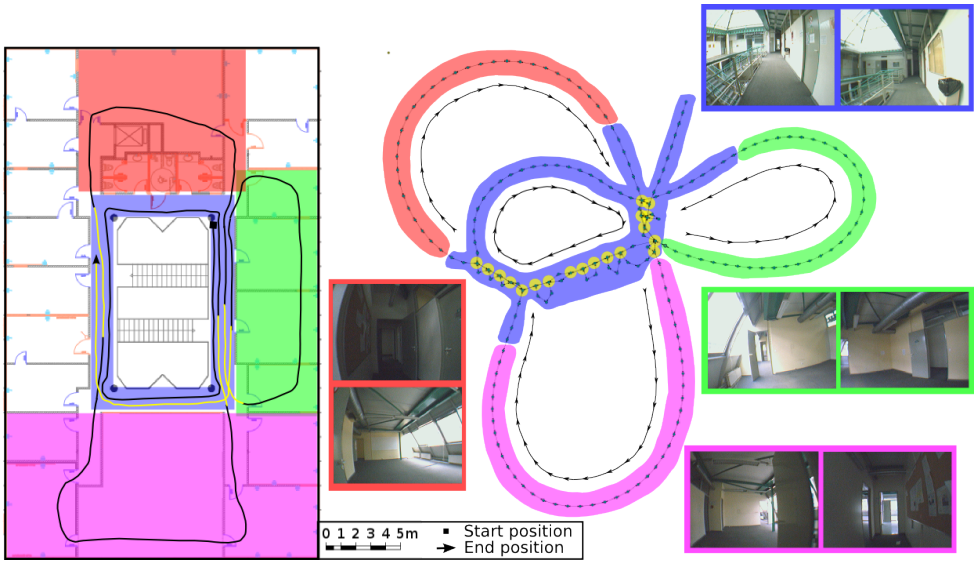

FIGURE 11.: Exemple de carte topologique créée à partir des détections de fermeture de boucles.

Enfin, à partir de ces détections de fermetures de boucles, il est possible de construire une carte topologique de l'environnement (Angeli et al., 2008b). Chaque noeud de la carte correspond à une position de l'environnement où une image a été prise et mémorise les mots visuels associés au travers de l'index inversé. Les cartes produites ne contiennent cependant pas d'information de position relative entre les noeuds et ne permettent donc pas d'obtenir la structure métrique de l'environnement (Figure 11).

En conclusion, ce modèle détecte de manière très robuste le retour d'une caméra dans une zone déjà visitée avec un taux de fausses alarmes très faible. Ces bonnes performances sont liées essentiellement aux qualités de la représentation des images par la méthode des sacs de mots visuels et au filtrage temporel apporté par le filtre Bayésien. Ce modèle permet également de réaliser une cartographie topologique, mais les cartes obtenues ne permettent cependant pas directement le guidage d'un robot.

2.1.5 Cartographie topo-métrique

Pour pouvoir guider un robot à partir d'une carte topologique, il est nécessaire de disposer de la capacité de passer d'un noeud de la carte à un autre. Une méthode possible est d'utiliser l'asservissement visuel entre les images associées à chaque noeud (Remazeilles and Chaumette, 2007), mais cela demande des traitements d'images relativement couteux et ne fonctionne plus en cas de perte de l'information visuelle. Une autre méthode, que nous avons choisie, est d'utiliser l'odométrie du robot qui permet d'estimer le déplacement métrique entre les deux positions.

L'inclusion de l'odométrie dans le modèle de cartographie topologique (section 2.1.4) a plusieurs avantages. Elle permet notamment de guider très simplement le robot sur une courte période en l'absence d'information visuelle. Elle permet également

Travaux réalisés dans le cadre du post-doctorat de Stéphane Bazeille, des stages de Zhenjiang Ni et Nicolas Beaufort et du projet VISIONAV avec Emmanuel Battesti (Bazeille and Filliat, 2010; Bazeille and Filliat, 2011).

d'améliorer l'étape de prédiction dans le filtre bayésien en prenant en compte exactement le déplacement du robot et ainsi d'améliorer la réactivité de la détection des fermetures de boucles. Nous avons pour cela utilisé un modèle standard (Thrun et al., 2005) supposant un bruit gaussien sur le déplacement en coordonnées polaires :

$$p(d,\alpha,\phi|d_u,\alpha_u,\phi_u) = G_{\mu_d,\sigma_d}(d-d_u)G_{\mu_\theta,\sigma_\theta}(\alpha-\alpha_u)G_{\mu_\phi,\sigma_\phi}(\phi-\phi_u) \tag{2.2}$$

où d,α est le déplacement du robot en coordonnées polaires dans le repaire de la position de départ et ϕ est le changement de direction du robot. $G_{\mu,\sigma}(X)$ est la gaussienne de moyenne μ et de variance σ^2.

L'ajout d'un algorithme de relaxation performant, TORO (Grisetti et al., 2007) permet enfin d'estimer assez précisément la structure géométrique de l'environnement et il devient possible d'estimer une trajectoire relative suffisamment précise pour rejoindre un but distant (Figure 12).

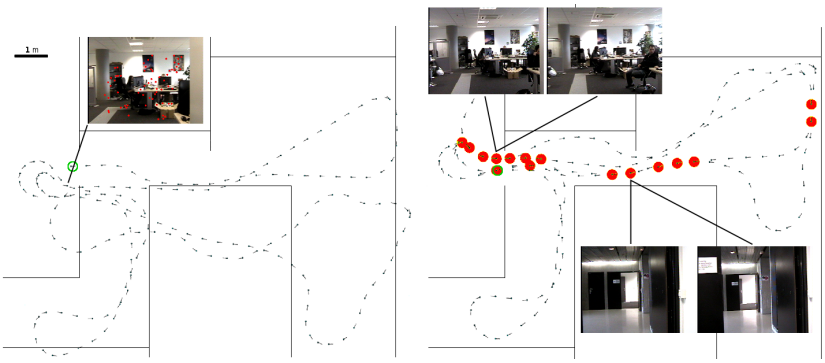

FIGURE 12.: Exemple de carte topo-métrique créée à partir de l'odométrie brute (à gauche) et après détections de fermeture de boucles et relaxation de la carte (à droite).

Dans le cadre du projet VISIONAV avec la société GOSTAI [1], nous avons réalisé une nouvelle implémentation de ce modèle dans le but de l'évaluer pour des robots disposant d'une puissance de calcul limitée. En effet, en utilisant une méthode de sacs de mots visuels statique (Nister and Stewenius, 2006), ce modèle permet de traiter une image de 320x240 pixels en moins de 30 ms. Ce traitement étant effectué en moyenne toutes les secondes, cela permet de ne mobiliser qu'une petite partie du processeur du robot pour la navigation. De plus, le guidage du robot étant réalisé par l'odométrie, le traitement des images peut se faire de manière totalement asynchrone, éventuellement sur un ordinateur distant avec des latences de communication importantes.

Ce modèle permet de réaliser des cartes topo-métriques fiables à partir de vision et d'odométrie. Il permet d'estimer correctement la structure des bâtiments et les cartes produites sont facilement utilisables pour guider un robot. La puissance de calcul nécessaire est de plus relativement limitée et les calculs peuvent être facilement déportés.

1. http://www.gostai.com

2.2 NAVIGATION MÉTRIQUE ET SÉMANTIQUE

Nous avons développé des approches de navigation métrique et sémantique dans le cadre de deux projets intégrés dont les objectifs étaient plus larges que la simple navigation.

2.2.1 Démonstrateur de robot de reconnaissance

Travaux réalisés au Centre Technique d'Arcueil en collaboration avec André Dalgalarrondo, Delphine Dufourd et l'équipe du groupe « Perception pour la Robotique » (Dalgalarrondo et al., 2004).

Afin d'illustrer les possibilités de la robotique dans un cadre militaire et les différents modes de contrôle possibles pour un robot mobile, nous avons implanté des capacités de navigation dans un démonstrateur. Ce démonstrateur intègre notamment un algorithme de cartographie basé sur la corrélation de scan laser (Röfer, 2002) et un filtre de Kalman pour la propagation des erreurs lors des fermetures de boucles. Il intègre également une méthode de planification utilisant l'algorithme A^* et une méthode de suivi de trajectoire et d'évitement d'obstacles dynamique utilisant la méthode de la fenêtre dynamique (Fox et al., 1997). L'ensemble de ces fonctionnalités est supporté par une architecture multi-agent hybride qui permet la gestion des différents modes de contrôle du robot (Dalgalarrondo, 2001).

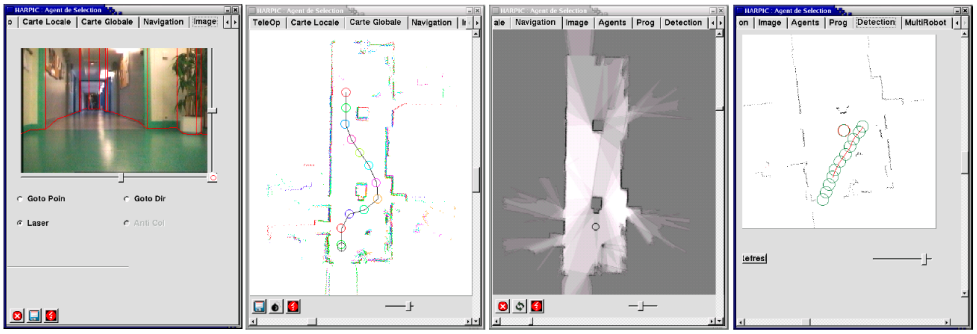

FIGURE 13.: Interface de contrôle du démonstrateur robotique. De gauche à droite : Téléopération par vision ; affichage de la cartographie ; déplacement planifié ; détection d'objets mobiles.

Une interface homme-machine développée sur un assistant personnel permet le contrôle du robot (Figure 13). Elle permet notamment de télé-opérer directement le robot à partir des données laser ou de l'image de la caméra, de lancer des comportements autonomes tels que le suivi de mur ou de couloir, et de réaliser des déplacements planifiés dans la carte de l'environnement qui a été construite. Un « module mission » simple permet également de détecter les personnes se déplaçant autour du robot à partir du télémètre laser (Dalgalarrondo et al., 2004).

Ce démonstrateur développé pour l'exploration de bâtiments bénéficie d'une méthode de navigation robuste et a permis d'étudier différents modes de supervision de la mobilité au travers d'une interface homme-machine dédiée.

2.2.2 Cartographie sémantique

Travaux réalisés dans le cadre du projet ANR PACOM, de la thèse d'Islem Jebari, et des stages de Marius Klein, Mathias Pointner, Hassène Tekaya et Guillaume Duceux (Jebari et al., 2011).

Depuis 2009, nous portons le projet PACOM dont le but est de participer au défi « CAROTTE [2] », une compétition de robotique organisée par l'Agence Nationale pour la Recherche et la Direction Générale pour l'Armement dont l'objet est de réaliser l'exploration et la cartographie autonome d'un bâtiment. Dans le cadre de ce projet, nous avons en particulier proposé de réaliser une cartographie sémantique contenant des informations de haut niveau qui sont directement intéressantes pour l'opérateur telles que les objets présents ou la structure des pièces (Galindo et al., 2005).

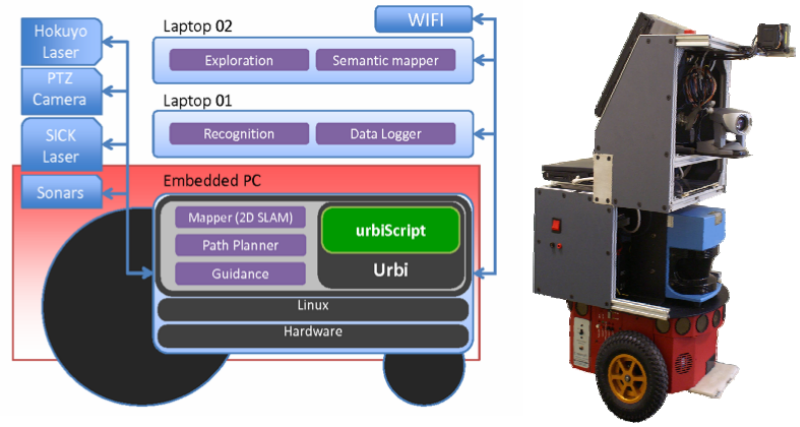

FIGURE 14.: Architecture logicielle basée sur Urbi (gauche) et matérielle (droite) du robot développé pour le projet PACOM.

Dans le cadre de ce projet, nous avons intégré une architecture logicielle complète, basée sur le logiciel Urbi de la société GOSTAI, intégrant des algorithmes de cartographie laser 2D, d'exploration, de détection visuelle d'objets et de cartographie sémantique (Figure 14). Pour la cartographie laser 2D, nous avons fait appel à une bibliothèque commerciale (Karto). L'algorithme de détection visuelle sera décrit dans le chapitre 3.

La cartographie sémantique intègre les différentes détections d'objets dans le repère global de la carte en utilisant un filtre de Kalman, classiquement utilisé pour la cartographie en robotique (Smith et al., 1988). Ceci permet d'avoir une estimation de la position de chaque objet et une évaluation de l'incertitude de cette estimation au travers des ellipses de covariance. Nous avons également intégré un algorithme de détection de pièces basé sur la détection des portes qui permet de segmenter l'environnement (Figure 15). Une carte 3D sous forme de nuage de points est générée à partir d'un second télémètre laser orienté verticalement.

Enfin, un algorithme d'exploration multi-objectifs permet d'assurer la cartographie complète de l'environnement par la télémétrie laser et de garantir que la recherche d'objet par vision a été exhaustive. Cet algorithme est un algorithme probabiliste qui cherche une position maximisant une somme de critères :

$$S_{pos} = \lambda_{vision} S_{vision} + \lambda_{map} S_{map} + \lambda_{dist} S_{dist}$$

2. http ://www.defi-carotte.fr

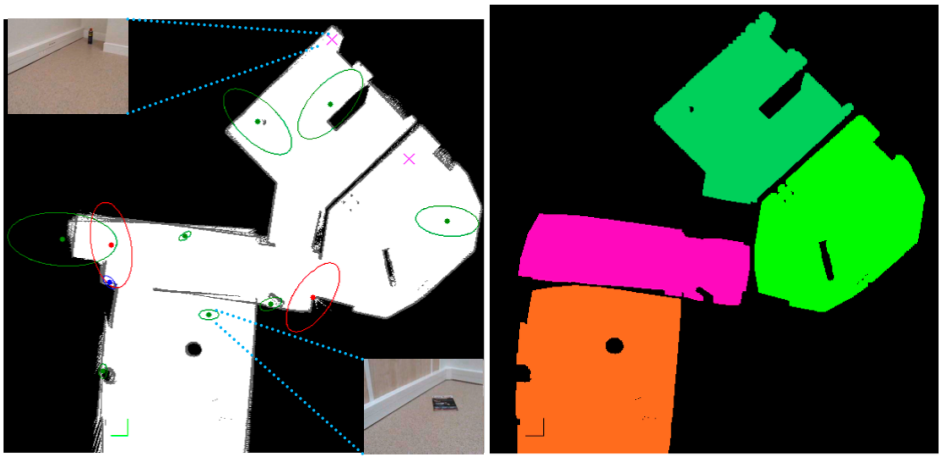

FIGURE 15.: Cartes sémantiques montrant les objets détectés dans l'environnement (gauche) et les pièces détectées (droite).

où S_{vision} est la surface non encore observée qui sera découverte par la caméra, S_{map} est la longueur des frontières qui seront observées par le laser (Yamauchi, 1998) et S_{dist} un score pénalisant la longueur du déplacement pour atteindre cette position. Cet algorithme réalise une exploration complète et relativement rapide de l'environnement.

Ce projet a permis de développer un système complet intégrant cartographie, navigation, exploration et détection d'objets. Lors de la première édition de la compétition, nous avons obtenu la troisième place. Deux autres compétitions avec des règles plus complexes sont prévues en 2011 et 2012.

3 PERCEPTION

Avec l'augmentation des performances des robots au niveau de la mobilité et des capacités de manipulation, les tâches que l'on peut imaginer leur confier deviennent de plus en plus complexes. Or pour pouvoir agir efficacement dans un environnement quotidien, il faut pouvoir percevoir un grand nombre d'éléments comme les obstacles au déplacement, les objets, les personnes. Malheureusement, les capacités de perception des robots en terme de versatilité et de robustesse restent encore limitées. Par exemple, en utilisant la vision uniquement, les meilleurs algorithmes actuels évalués lors du « Visual Object Classes Challenge » sont capables de détecter environ 20% des chaises présentes dans des images (Everingham, 2010).

Un autre point limitant est le niveau d'abstraction des informations extraites des perceptions. Ainsi, s'il est relativement simple de détecter des obstacles à l'aide d'un télémètre laser, il serait plus intéressant d'avoir des informations sur la possibilité de pousser cet obstacle par exemple, ce qui passe par une identification beaucoup plus précise de l'obstacle et des connaissances de plus haut niveau de ses caractéristiques. D'une manière générale, il est donc intéressant de pouvoir fournir au système des informations ayant un fort contenu *sémantique*.

Enfin, au delà de l'identification d'un élément particulier, il est souvent intéressant de disposer d'information de catégorie. En robotique, un type de catégorie particulièrement intéressant est lié à la notion d'affordances (Gibson, 1986) qui suppose que l'homme est capable de reconnaitre directement des catégories d'objets par les actions qu'il est possible de réaliser dessus. Par exemple, l'homme paraît capable de détecter les objets « que l'on peut prendre ». L'information de catégorie peut également fournir un contexte qui sera nécessaire pour interpréter des commandes ou exécuter des tâches, par exemple en reconnaissant que telle pièce est une cuisine pour chercher de la nourriture.

Au niveau des capteurs, le plus utilisé pour la navigation reste le télémètre laser grâce à la précision et à la fiabilité de ses mesures. Toutefois, la vision est très intéressante grâce au faible coût des caméras et à la grande quantité d'information contenue dans les images. Dans nos travaux, nous avons ainsi utilisé ces deux capteurs, soit isolément, soit en combinaison pour la navigation ou la reconnaissance d'objets. Nous avons cherché d'une part à fournir des représentations bas-niveau fiables pour la robotique et d'autre part à augmenter le niveau sémantique des informations produites.

3.1 PERCEPTION POUR LA ROBOTIQUE

3.1.1 Problème de la représentation

Pour traiter une information dans n'importe quel système automatisé, il faut pouvoir la représenter. Or il est connu depuis longtemps que la représentation choisie influe beaucoup sur la difficulté de traitement de cette information par la suite, que ce soit en vision ou plus généralement en intelligence artificielle (Davis et al., 1993).

En vision, ce problème de la représentation est particulièrement important. Les cameras fournissent un volume de données important qui doit en général être réduit pour pouvoir être traité par des algorithmes d'apprentissage ou de reconnaissance. Ces données sont de plus entachées de bruits de nombreuses sources différentes comme le bruit de mesure de la caméra en basse lumière ou encore la variation des conditions environnementales (passage du jour à la nuit par exemple).

Parmi les nombreuses représentations existantes, nous nous sommes intéressés aux représentations à base de caractéristiques locales qui peuvent offrir une bonne robustesse aux occultations et aux changements de points de vue. En particulier, nous avons utilisé le modèle de *sacs de mots visuels* (Sivic and Zisserman, 2003) qui permet de représenter une image comme une collection non ordonnée de caractéristiques locales (voir section 3.1.3). Au delà de ces représentations très bas niveau, nous nous sommes également intéressés à des représentations plus complexes pouvant à terme servir de base à des fonctions cognitives plus proches de l'intelligence artificielle (voir section 3.1.4).

3.1.2 Caractérisations bas-niveau

Il existe de nombreuses caractérisations locales bas-niveau des images utilisables pour les tâches de reconnaissance de lieux ou d'objets. Nous pouvons en citer quatre grands types :

- Les descriptions denses qui sont calculées en chaque point de l'image, par exemple via l'application d'un banc de filtre, tels que les filtres de Gabor (Daugman, 1988). Elles ont souvent l'inconvénient d'être lourdes en calcul et donc plus difficilement applicables en robotique.
- Les points d'intérêts qui représentent des points particuliers de l'image tels que les coins (Harris and Stephens, 1988) associés à des descripteurs. Ces dernières années des détecteurs invariants aux changements d'échelle et à la rotation sont devenus très populaires de par leur robustesse et leurs performances. Dans nos travaux, nous avons notamment utilisé SIFT (Lowe, 2004) et SURF (Bay et al., 2006).
- Les contours de l'image qui représentent, entre autre, les bords des objets. Ces caractéristiques sont par exemple bien adaptées à certaines catégories d'objets (Ferrari et al., 2008) ou à la manipulation d'objets (Krüger and Wörgötter, 2004; Kraft et al., 2010).
- Les surfaces qui permettent de segmenter l'image en un ensemble des régions uniformes qui sont souvent chacune associée à un objet unique. En particulier,

des approches utilisant des *superpixels* sont particulièrement intéressantes pour des tâches de labelisation sémantique qui cherchent à associer une information à chaque pixel de l'image (Ren and Malik, 2003).

Ces caractérisations fournissent des informations très différentes, qui peuvent être spécifiquement adaptées à une tâche, ou au contraire pertinentes pour de nombreuses tâches comme par exemple dans l'approche « Early Cognitive Vision » (Krüger et al., 2010). Dans nos travaux, nous avons principalement utilisé des caractéristiques génériques existantes : SIFT (Lowe, 2004), SURF (Bay et al., 2006) ainsi que des histogrammes de teinte locaux permettant de prendre en compte les couleurs (Filliat, 2007; Angeli et al., 2008a) et des superpixels caractérisés par leur texture et leur couleur (Micusik and Kosecka, 2009).

3.1.3 Sacs de mots visuels incrémentaux

FIGURE 16.: Illustration du codage sous forme de « sac de mots visuels ».

Sur la base de ces représentations de bas niveau, nous avons utilisé un modèle appelé « sacs de mot visuels » (Sivic and Zisserman, 2003) dans plusieurs applications (sections 2.1.3, 2.1.4, 2.1.5, 3.2.1). Ce terme provient de méthodes de classification de textes qui représentent un texte comme un simple « sac » de ses mots, en en oubliant la structure et la grammaire. Par analogie, ce modèle peut être utilisé pour représenter des images en prenant comme « mots » des caractéristiques locales des images quantifiées selon un dictionnaire (Figure 16) et dont on mémorise simplement un histogramme des occurrences.

Au lieu de construire au préalable le dictionnaire sur une base de données d'images comme dans la plupart des applications (e.g., (Nister and Stewenius, 2006)), nous avons développé une méthode de construction incrémentale de dictionnaire (Filliat, 2007) qui permet de commencer avec un dictionnaire vide et de le construire au fur et à mesure de la découverte de l'environnement. Notre modèle permet donc de ne pas faire d'hypothèse *a priori* sur le type d'environnement auquel il devra faire face.

Les mots dans notre modèle sont des boules de rayon fixe dans l'espace des caractéristique locales. La construction du dictionnaire consiste simplement à ajouter un mot nouveau centré sur chaque caractéristique qui n'appartient pas à un mot déjà existant. La taille des boules est appelée le rayon du dictionnaire et influe sur la taille du dictionnaire, les performances des algorithmes et les temps de calcul (Filliat, 2007).

Lors de l'utilisation des méthodes de sac de mots avec des vocabulaires de grande taille comme cela se fait dans notre méthode, la recherche du mot correspondant à une caractéristique locale est un processus coûteux en temps de calcul. Nous avons donc

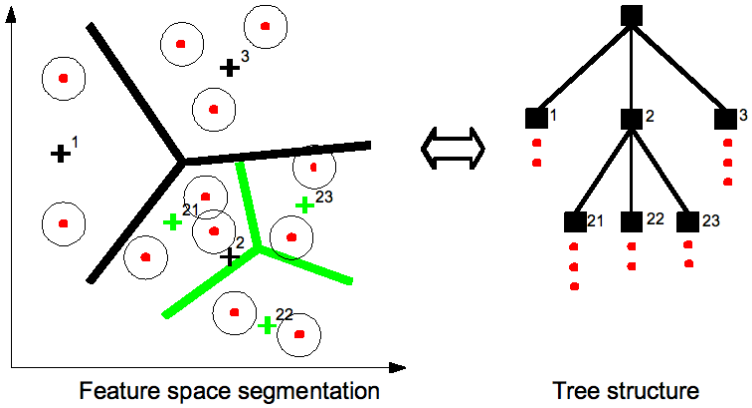

FIGURE 17.: Illustration de l'arborescence des dictionnaires avec $k = 3$. A gauche : Les croix illustrent les centres des noeuds, les cercles illustrent les mots. Droite : Seuls les centres des mots sont stockés dans les feuilles de l'arbre.

développé une structure arborescente du dictionnaire pour accélérer cette opération. Cette structure est similaire à celle de (Nister and Stewenius, 2006), mais construite incrémentalement (figure 17). Chaque noeud interne de l'arbre a ainsi un ensemble k de fils, chacun étant défini par un centre dans l'espace des caractéristiques. Chaque fils enregistre les mots dont les centres sont les plus proches de son centre.

Le processus de construction commence simplement avec un noeud racine vide. Tout nouveau mot qui doit être ajouté au dictionnaire est directement ajouté au noeud feuille auquel appartient son centre. Si le nombre de mots stockés dans cette feuille est au-dessus d'un seuil n_w, la feuille est divisée en k fils. Les centres des fils sont définis par l'application de l'algorithme des *k-moyennes* (MacQueen, 1967) aux centre des n_w mots.

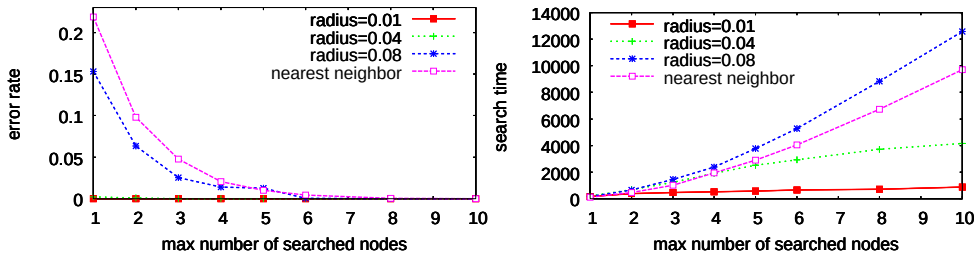

FIGURE 18.: Gauche : Taux d'erreur pour la recherche du mot correspondant à un point SIFT en fonction du nombre de fils exploré pour chaque noeud. Droite : Temps de recherche correspondant pour un dictionnaire de 15000 mots construit avec $k = 10$.

Comme le montrent Beis et Lowe (Beis and Lowe, 1997), chercher des vecteurs dans ce type de structures en grande dimension (par exemple 128 pour les descripteurs SIFT) conduit à une complexité similaire ou supérieure que celle de la recherche linéaire naïve parce qu'un grand nombre de noeuds est examiné, compromettant ainsi tout intérêt de l'utilisation d'une structure arborescente. Ce problème a été résolu dans (Beis and Lowe, 1997) dans le cas de *kd-trees* par une procédure de recherche approximative rapide. Nous avons utilisé un procédé similaire, en limitant le nombre

de fils à explorer dans chaque noeud et en cherchant en priorité dans les fils dont les frontières sont les plus proches de la caractéristique recherchée. Cette procédure permet une recherche très rapide - au prix d'un faible pourcentage d'erreurs (Figure 18).

Cette méthode incrémentale appliquée pour la détection de fermeture de boucles permet ainsi de traiter sans adaptation des environnements intérieurs, extérieurs ou mixtes (Angeli, 2008). Il est intéressant de noter que nous avons également appliqué avec succès ce modèle à la reconnaissance sonore de mots (voir section 3.2.2).

3.1.4 Représentations intermédiaires

Travaux réalisés lors du stage de Roland Goffettre et de la thèse de Natalia Lyubova dans le cadre du projet MACSi.

Les caractéristiques bas-niveau peuvent servir directement de support à de nombreux algorithmes de perception, mais leur pouvoir de discrimination prises isolément ne permet pas en général d'identifier directement un objet ou un élément de l'environnement. Ainsi dans la plupart des algorithmes, comme dans le modèle de sacs de mots visuels présenté précédemment, un ensemble de ces caractéristiques est utilisé pour reconnaitre un objet.

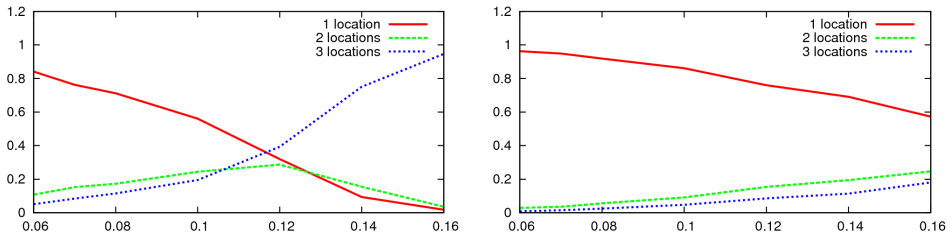

FIGURE 19.: Exemple de capacité de discrimination de points d'intérêt SIFT (à gauche) et de paires de points SIFT (à droite) en fonction de la taille du dictionnaire de mots visuels (voir section 3.1.3) pour une tâche de reconnaissance de pièces (voir section 2.1.3). Les courbes indiquent le pourcentage d'éléments vus dans 1, 2 ou 3 pièces.

Il est également possible d'utiliser des représentations intermédiaires plus complexes que ces caractéristiques de bas-niveau. Par exemple, des structures locales de points d'intérêts (Quack et al., 2006) seront plus discriminantes que les caractéristiques isolées. Nous avons ainsi développé une approche utilisant des paires de points d'intérêts voisins que nous avons évaluée sur des tâches de reconnaissance d'objets et de pièces (Goffettre, 2007). Le pouvoir de discrimination de ces paires est effectivement supérieur (Figure 19), mais elles n'ont pas permis d'augmenter les performances brutes de reconnaissance d'objets. Cette approche permet néanmoins de diminuer les temps de calcul en conservant les mêmes performances avec un dictionnaire de caractéristiques de base de plus petite taille (voir section 3.1.3).

Nous verrons également dans la section 3.2.2 que l'utilisation d'une étape intermédiaire de classification non supervisée des histogrammes d'occurrence de mots visuels peut fournir une représentation plus adaptée pour certains problèmes. Cette technique d'utilisation hiérarchique d'apprentissage non supervisé se rapproche des méthodes de *Deep Learning* (Arel et al., 2010). Ces méthodes utilisent des architec-

tures récursives, employant plusieurs fois le même algorithme d'apprentissage pour permettre la découverte non supervisée de caractéristiques intermédiaires pertinentes des données.

Au delà de ces premières approches, il est important de pouvoir gérer des représentations de complexité croissante pour passer de l'information bas-niveau à une information directement pertinente pour les processus haut-niveau tels que l'interaction avec l'homme ou la navigation. Il existe une très grande variété d'approches possibles pour cela, et l'inspiration biologique peut être une guide intéressant. Dans le cadre de la robotique développementale (voir section 4.2), ces étapes de représentations intermédiaires se fondent en général sur le concept de *proto-objets*, décrits en psychologie cognitive (Pylyshyn, 2001), qui sert d'intermédiaire entre l'information visuelle au niveau de la rétine et les objets reconnus qui sont utilisés par les processus cognitifs plus complexes. Plusieurs modèles de proto-objets utilisant différentes approches de traitement d'images (Kraft et al., 2008; Orabona et al., 2008) ont été appliqués à des robots pour apprendre à reconnaître les éléments de leur environnement. Nous développons actuellement de nouvelles approches utilisant des structures locales de points d'intérêt dans le cadre du projet MACSi (voir section 5.2.1).

3.1.5 Perception active

La perception dans le domaine de la robotique peut tirer avantage des capacités d'action du robot afin de simplifier les problèmes posés. Ce domaine de la *perception active* est une différence essentielle entre les approches de vision par ordinateur pures et les approches de vision pour la robotique. Ainsi des méthodes de perception assez simples, mises en oeuvre dans le cadre de la robotique peuvent permettre d'obtenir de très bonnes performances.

Durant nos travaux de thèse (voir section 2.1.2), nous avons mis en oeuvre une méthode de perception active pour permettre la localisation de notre robot à l'aide d'une caméra directionnelle en remplacement d'une caméra panoramique (Filliat and Meyer, 2000; Filliat, 2001). Cette méthode permet d'une part de cartographier et d'autre part d'améliorer la localisation. Ainsi lorsque le robot est dans une zone inconnue ou si sa localisation est imprécise, la caméra est dirigée dans une direction qui permet d'améliorer l'estimation de la position. Lorsque la localisation du robot est précise au contraire, la caméra est dirigée dans une direction pour laquelle la carte est incomplète afin d'améliorer la représentation de l'environnement.

Pour le choix de la direction de la caméra, nous avons testé différentes stratégies reposant sur la quantité d'information disponible dans chaque direction ou sur le gain d'entropie de la probabilité de position du robot après intégration de l'information visuelle (Roy and Thrun, 1999). Cette dernière méthode se révèle très efficace lorsque la carte de l'environnement est complète avec des performances proches de l'utilisation d'une caméra panoramique. Par contre, durant la cartographie, l'incomplétude de la carte la rend complètement inefficace, et la première méthode plus simple devient beaucoup plus efficace (Filliat, 2001).

Nous avons également utilisé une méthode de perception active pour notre modèle de navigation qualitative (section 2.1.3). La méthode de vote utilisée pour prédire la

pièce courante ou la direction du but permet en effet d'intégrer plusieurs images afin d'augmenter la qualité des décisions. La décision d'ajouter des images se base sur la qualité du vote : si le résultat du vote est ambigu, une nouvelle image est prise en compte. Comme le modèle du monde n'est pas structuré géométriquement (suite à l'utilisation de l'approche des sacs de mots), la nouvelle image est prise depuis la même position du robot, mais avec une direction de caméra aléatoire. Cette stratégie permet notamment de gérer les images non informatives (par exemple des images de sol si le sol est le même dans toutes les pièces) ou les élément dynamiques tels qu'une personne présente devant le robot (Filliat, 2008).

3.2 APPLICATIONS DÉVELOPPÉES

3.2.1 Reconnaissance d'objets

Dans le cadre du projet PACOM, nous avons développé une méthode de détection d'objets afin de les mémoriser dans la carte sémantique (section 2.2.2). Nous avons pour cela utilisé un modèle de sacs de mots visuels utilisant un dictionnaire statique (Nister and Stewenius, 2006). Pour cette application, nous avons pu créer à l'avance une base de donnée d'images des objets à reconnaitre.

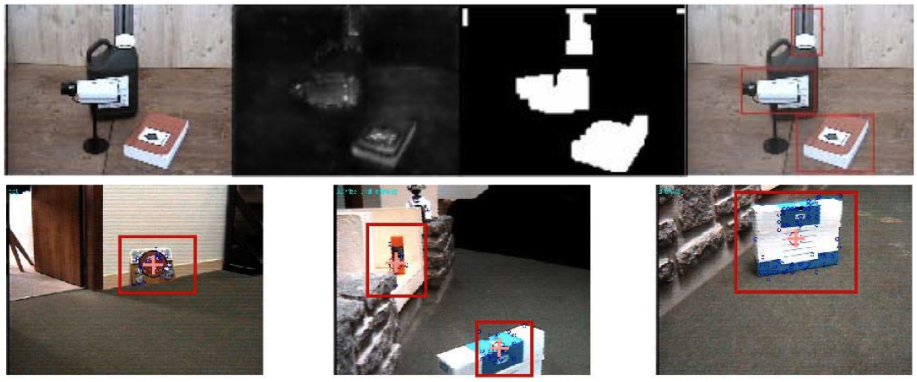

FIGURE 20.: Haut : Processus de segmentation des images en objets candidats. La saillance visuelle est calculée pour toute l'image, puis seuillée et les composantes connexes obtenues correspondent aux objets possibles. Bas : Exemple d'objets reconnus lors de l'exploration d'un bâtiment.

Travaux réalisés durant le stage et la thèse de Cédric Meyer dans le cadre du projet PACOM et avec Pierre Rouanet et Pierre-Yves Oudeyer dans le cadre de l'équipe INRIA FLOWERS (Jebari et al., 2011; Rouanet et al., 2010)

La méthode de reconnaissance d'objet est une méthode de vote très similaire à la méthode de reconnaissance de pièces présentée dans la section 2.1.3. Pour l'apprentissage, tous les mots visuels extraits des images de référence sont associés à l'objet contenu dans l'image. Pour la prédiction, à partir de l'image courante, les mots visuels extraits votent pour les objets avec lesquels ils ont été associés. Un objet est alors reconnu si le vote atteint une qualité suffisante. Cependant, contrairement à la reconnaissance de pièce, il peut y avoir plusieurs objets différents dans une même image. Pour permettre la reconnaissance de ces différents objets, nous avons segmenté l'image en objets candidats à partir d'une méthode de saillance visuelle (Butko

et al., 2008). Chaque zone saillante est alors analysée pour déterminer si elle contient ou non un objet (Figure 20).

Afin de pouvoir détecter une large palette d'objets texturés ou colorés, nous avons utilisé d'une part des points d'intérêt SURF (Bay et al., 2006) et d'autre part des histogrammes de couleur locaux (Filliat, 2007). Nous avons également intégré une estimation de la distance des objets afin d'estimer leur position dans la carte. Pour les objets détectés à l'aide des points SURF, cette distance est estimée à l'aide de la variation d'échelle entre l'objet détecté et l'image d'apprentissage pour laquelle la distance de l'objet est connue. Pour les objets colorés, cette distance est estimée en faisant l'hypothèse d'un objet posé au sol (Jebari et al., 2011).

En collaboration avec Pierre Rouanet et Pierre-Yves Oudeyer de l'équipe INRIA FLOWERS, nous avons également utilisé une méthode très similaire dans le cadre d'une étude d'interfaces permettant de faire apprendre des nouveaux objets à un robot (Rouanet et al., 2010) (voir section 3.1.5).

3.2.2 Reconnaissance audio-visuelle

Dans le cadre de notre collaboration avec l'équipe INRIA FLOWERS, nous avons commencé à explorer des méthodes de perception intégrant l'information sonore. Ces travaux se placent dans le cadre de la robotique développementale (voir section 4.2) où nous cherchons à mettre en place des méthodes d'apprentissage dans un contexte d'interaction homme-robot, en évitant par exemple l'utilisation de bases d'apprentissage annotées de manière très précise.

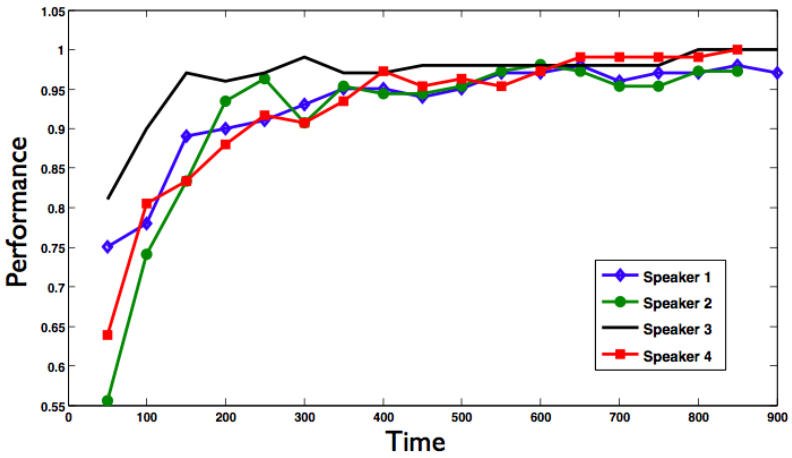

FIGURE 21.: Performance de la reconnaissance sonore de mots en fonction du nombre d'exemples d'apprentissage prononcés par quatre locuteurs.

Travaux réalisés en collaboration avec Olivier Mangin, Pierre-Yves Oudeyer et Louis ten Bosch (Mangin et al., 2010; ten Bosch et al., 2011).

Dans un premier temps, nous avons appliqué notre modèle de sacs de mots visuels incrémentaux à la reconnaissance sonore. L'objectif était de permettre un apprentissage incrémental de la reconnaissance de mots à partir du signal sonore brut sans introduire aucun *a priori* sur la phonétique, et en utilisant simplement un ensemble de phrases contenant ces mots. Pour cela, nous avons redéfini les caractéristiques de base extraites du signal. En remplacement des points d'intérêts issus de l'image, nous

avons utilisé des descripteurs classiques en traitement du son (Mel-Frequency Cepstral Coefficients (MFCC) and Relative Spectral Transform - Perceptual Linear Prediction (RASTA-PLP) (Ellis, 2005)) calculés sur des fenêtres temporelles glissantes extraites du signal sonore. A partir de ces caractéristiques, nous avons utilisé la méthode de sacs de mots incrémentaux (section 3.1.3) et une méthode de vote similaire à celle utilisée pour la reconnaissance d'objets (section 3.2.1).

Nous avons appliqué ce modèle à une base de données sonores contenant des phrases prononcées par quatre locuteurs différents. L'objectif était de prédire si un mot particulier apparaissait dans une phrase donnée, sans chercher à segmenter le mot au sein de la phrase. Étant donné la simplicité et la généricité du modèle, les performances obtenues sont très bonnes (Figure 21), proches des performances obtenues avec des méthodes plus complexes (ten Bosch et al., 2008).

Dans le cadre d'une collaboration avec Louis ten Bosch de l'Université Radboud de Nijmegen et avec l'équipe INRIA FLOWERS, nous avons également développé une méthode permettant de découvrir de manière non supervisée le nom et l'apparence visuelle associés à un objet. Cette méthode utilise également une représentation à base de sacs de mots visuels et sonores, associée à un algorithme non supervisé appelé *Non negative Matrix Factorization (NMF)* qui est décrit dans le prochain chapitre consacré à l'apprentissage (section 4.1.2). L'objectif est de reproduire fonctionellement ce que fait un petit enfant qui est capable de retrouver un objet nommé par ses parents bien avant de savoir parler, sans aucune supervision explicite. Dans l'expérience réalisée, le système apprend en percevant un objet et une phrase contenant le nom de l'objet simultanément. Après apprentissage, une phrase contenant le nom d'un objet est fournie au système qui doit choisir le bon objet parmi plusieurs qui lui sont présentés visuellement.

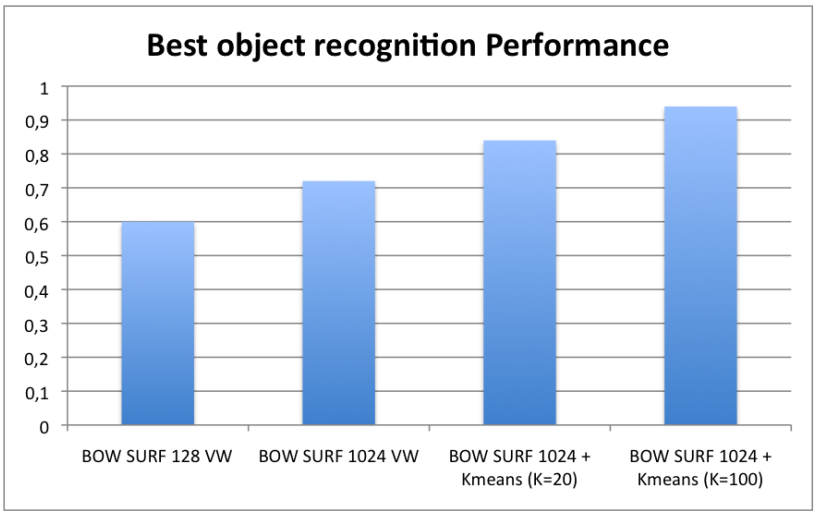

FIGURE 22.: Performances de reconnaissance d'objets en fonction de la représentation visuelle utilisée : Sacs de mots visuels avec un dictionnaire de 128 ou 1024 mots et Sacs de mots visuels avec une catégorisation non supervisée des images par k-means (k=20 et 100).

Dans cette application, la représentation à base de sacs de mots visuels ne permet cependant pas d'obtenir des performances satisfaisantes, et il a été nécessaire d'in-

troduire une phase de catégorisation non supervisée intermédiaire pour l'information visuelle. Les histogrammes d'occurrence de mots visuels représentant les images sont ainsi eux-même catégorisés pour construire une base de représentation dans laquelle les nouvelles images seront projetées. Cette catégorisation peut se réaliser avec un algorithme de type *k-mean*, ou hiérarchiquement en appliquant une première fois la méthode NMF à la seule partie visuelle. Cette nouvelle représentation plus compacte permet d'améliorer sensiblement les performances (Figure 22). Ces résultats montrent concrètement l'intérêt d'utiliser des représentations de niveau intermédiaire comme support de fonctions de haut-niveau (voir section 3.1.1).

3.2.3 Segmentation sémantique

Travaux réalisés en collaboration avec Fabio Gonzalez et Antoine Manzanera et dans le cadre de la thèse d'Islem Jebari et du projet PACOM (Gonzalez, 2010).

Les capacités de perception *sémantique* mentionnées en introduction de ce chapitre doivent permettre de reconnaitre un grand nombre d'éléments d'apparence et de taille très différentes. Détecter tous ces éléments individuellement peut s'avérer difficile et il peut être intéressant de prendre en compte une information de *contexte* au sein de l'image pour améliorer les performances (Galleguillos and Belongie, 2010).

FIGURE 23.: Exemple de résultats obtenus par la méthode de segmentation sémantique sur la base de données « CamVid » (Brostow et al., 2008). Labels : Gris : Ciel, Rose : route, Bleu : Trottoir, Rouge : Batiment, Violet : Véhicules

Dans cette optique, nous nous sommes intéressés à la labelisation sémantique de scènes perçues depuis un véhicule dans un contexte urbain lors du séjour sabbatique à l'ENSTA ParisTech de Fabio Gonzalez (Gonzalez, 2010). L'objectif est d'associer un label (route, bâtiment, véhicule, piéton...) à chaque pixel de l'image. Nous avons utilisé pour ces travaux le « Cambridge-driving Labeled Video Database (CamVid) » (Brostow et al., 2008).

A cette occasion, nous avons implémenté une segmentation des images à base de superpixels (Micusik and Kosecka, 2009), chaque superpixel étant représenté par un descripteur SIFT et par sa couleur moyenne. Ce descripteur est ensuite quantifié au travers d'un dictionnaire qui lui associe un « mot visuel ». A partir de ces mots visuels, le label associé à chacun des superpixels est estimé par un modèle de Champs de Markov prenant en compte à la fois l'apparence du superpixel, des informations sur sa position géométrique dans l'image et les dépendances des labels entre superpixels

voisins. Ces différents termes sont modélisés sous forme d'une fonction d'énergie que l'on cherche à minimiser :

$$E(L) = \alpha E_{app}(L) + \delta E_{geom}(L) + \beta E_{edge}(L) + \gamma E_{prior}(L), \tag{3.1}$$

avec :

- $E_{app}(L) = -\sum_{l_i \in V} \log P(x_i^{app} | l_i)$
- $E_{geom}(L) = -\sum_{l_i \in V} \log P(x_i^{geom} | l_i)$
- $E_{edge}(L) = -\sum_{(i,j) \in Ed} \log P(l_i, l_j)$
- $E_{prior}(L) = -\sum_{l_i \in V} \log P(l_i)$

où $L = (l_1, \ldots l_i \ldots)$ est l'ensemble des labels des superpixels de l'image, x_i est le superpixel i, x_i^{app} est son apparence codée par son descripteur associé, x_i^{geom} sa position dans l'image et V l'ensemble de ses superpixels voisins. E_{app} correspond ainsi à la probabilité du label l_i connaissant l'apparence du superpixel x_i, E_{geom} correspond à la probabilité (modélisée par une gaussienne) du label l_i connaissant sa position dans l'image, E_{edge} correspond à la probabilité de trouver le label l_i dans le voisinage du label l_j et E_{prior} correspond à un *a priori* sur les probabilités des labels. Toutes ces données sont apprises sur la base d'apprentissage CamVid qui a été annotée à la main. Ce coût est maximisé grâce à l'algorithme Max-sum (Werner, 2007).

	bicyclist	building	car	pedestrian	road	sidewalk	tree	Average
MRF Appearance	9.8%	52.5%	37.7%	15.6%	84.5%	22.8%	39.4%	33.29%
MRF Geometric + Appear.	14.3%	55.4%	53.9%	34.50%	84.8%	55.1%	37.8%	43.09%
(Micusik and Kosecka, 2009)	15%	61.9%	72.4%	56.5%	89.6%	83.1%	60%	52.2%

FIGURE 24.: Précision pour quelques classes et précision moyenne obtenue pour notre approche avec et sans information géométrique comparées aux meilleurs résultats de (Micusik and Kosecka, 2009).

Cette première approche relativement simple, notamment au niveau de l'information géométrique utilisée, a donné des résultats prometteurs (Figures 23 et 24). Nous l'étendons actuellement dans le cadre de la cartographie sémantique multi-capteurs (section 2.2.2) à l'utilisation combinée de données télémétriques et visuelles pour la labelisation sémantique en intérieur.

4 APPRENTISSAGE

Après les robots utilitaires spécialisés, des robots ludiques tels que Nao d'Aldebaran Robotics [1] arrivent sur le marché. Ces modèles que l'on peut qualifier de « compagnons » ont un rôle social et un intérêt essentiellement dans le cadre des interactions qu'ils peuvent engager avec leurs utilisateurs. L'un des challenges importants pour ces robots est alors de fournir des interactions riches et renouvelées afin d'intéresser leur propriétaire. A moyen terme, des versions plus évoluées de ces robots devraient pouvoir améliorer la qualité de vie de personnes âgées ou dépendantes (Tapus et al., 2007). Ils permettront à ces personnes de rester autonomes quelques années supplémentaires en assurant une forme de surveillance et d'assistance pour des gestes simples du quotidien.

Pour ces robots domestiques, l'aspect social, la capacité d'interaction avec les humains, est donc primordial (Chatila, 2008). Ceci se traduit d'un point de vue mécanique par une autonomie de déplacement, la capacité de saisir, de manipuler des objets tout en garantissant la sécurité des utilisateurs, mais pose surtout des défis du point de vue « cognitif ». Ainsi ces robots doivent intégrer des capacités de perception et d'interprétation des situations largement supérieures à ce qui est possible aujourd'hui. Ils doivent par exemple pouvoir détecter des objets, des visages, interpréter les expressions et les gestes de l'humain. D'une manière générale, ils se trouvent contraints de « comprendre » les situations complexes caractéristiques de l'environnement quotidien des humains. De plus ces capacités doivent être évolutives : ils doivent apprendre à reconnaître de nouveaux objets, de nouvelles personnes, apprendre à jouer à un nouveaux jeux, raconter de nouvelles histoires. Dans ce contexte, doter ces robots de capacités d'apprentissage est donc inévitable.

De très nombreux travaux de robotique, en particulier pour l'aspect perception, reposent d'ores et déjà sur des techniques d'apprentissage. Cependant, ces techniques restent souvent lourdes à mettre en oeuvre et il reste beaucoup à faire pour rendre cet apprentissage simple et intuitif pour les utilisateurs non spécialistes. Dans nos travaux, nous avons donc utilisé différentes méthodes d'apprentissage incrémentales qui sont bien adaptées à l'apprentissage en ligne, et nous avons cherché à rendre ces méthodes simples à utiliser dans un cadre d'apprentissage interactif. En particulier, dans le cadre de l'approche développementale de la robotique, nous cherchons à développer des méthodes permettant d'apprendre des capacités de perception dans un cadre d'interaction avec des humains.

1. http ://www.aldebaran-robotics.com/

4.1 TECHNIQUES MISES EN OEUVRE

Dans les modèles que nous avons développé, nous avons eu besoin de méthodes d'apprentissage incrémentales qui permettaient de mettre à jour les modèles rapidement, notamment lors d'une interaction avec un humain. Nous n'avons pas développé de méthodes nouvelles pour cela, mais avons utilisé plusieurs méthodes classiques adaptées à ces problématiques.

4.1.1 Méthodes Bayésiennes

Nous avons utilisé des méthodes Bayésiennes naïves (Rish, 2001) dans plusieurs de nos applications : navigation qualitative (section 2.1.3), détection de fermeture de boucles (section 2.1.4) et reconnaissance d'objets (section 3.2.1). Ce sont des méthodes génératives qui permettent d'estimer la probabilité de plusieurs catégories, par opposition à des méthodes discriminatives telles que les « Séparateurs à Vastes Marges » (Cortes and Vapnik, 1995) qui cherchent à décider si un example appartient ou non à chaque catégorie (Ng and Jordan, 2001).

Dans nos applications, ces méthodes permettent d'évaluer la probabilité de la classe correspondant à une image, en supposant les éléments de la perception indépendants :

$$P(C|Z) = \eta P(Z|C) = \eta \prod P(z_i|C)$$

où C est la classe recherchée (par exemple l'objet à reconnaitre), Z la perception réalisée (l'image) et z_i des caractéristiques élémentaires de Z (les mots visuels extraits de l'image). Bien que l'hypothèse d'indépendance ne soit en général pas vérifiée, cette méthode donne néanmoins souvent de très bons résultats dans la pratique.

Le principal avantage de ces méthodes pour la robotique est la simplicité de la mise à jour du modèle. En effet, lors de l'ajout d'un exemple d'apprentissage, il suffit de mettre à jour les probabilités $P(z_i|C)$, ce qui peut se faire simplement en maintenant un décompte des occurrences de l'élément z_i pour la classe C. De même, il est possible d'ajouter simplement une nouvelle catégorie à tout moment sans apprentissage complexe. Comme nous l'avons fait dans la plupart de nos applications, il est possible d'implémenter cette méthode de façon très efficace à l'aide d'un index inversé qui relie chaque élément z_i aux classes pour lesquelles il a été perçu. Cet index permet alors d'implémenter une méthode de vote qui est équivalente à cette méthode bayésienne naïve.

4.1.2 Apprentissage non supervisé

Travaux réalisés en collaboration avec Pierre-Yves Oudeyer et Louis ten Bosch (ten Bosch et al., 2011).

Dans le cadre d'une collaboration avec Louis ten Bosch et l'équipe INRIA FLOWERS, nous avons également développé une méthode permettant de découvrir de manière non supervisée le nom et l'apparence visuelle associés à un objet (voir section 3.2.2 pour la description de la tâche).

Pour cela, nous avons utilisé une méthode de factorisation appelée *Non negative Matrix Factorization (NMF)* (Hoyer and Dayan, 2004). Cette méthode permet de trouver une base de représentation d'un ensemble de données dans laquelle elles sont

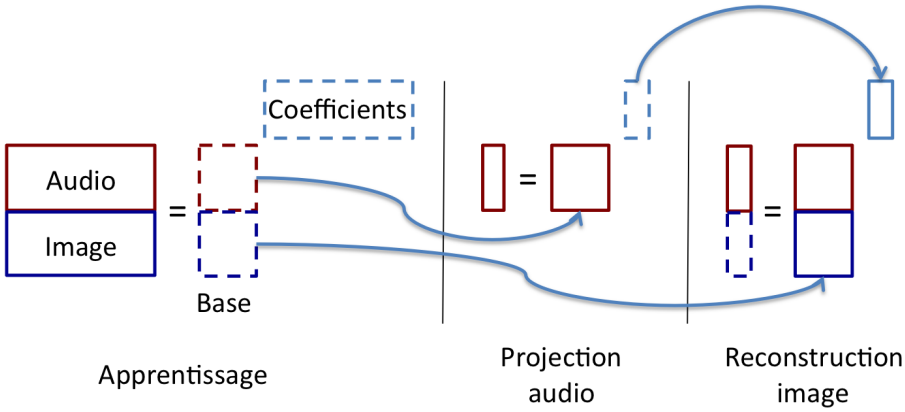

FIGURE 25.: Illustration de la méthode NMF employée. Les données sont en traits plein, les résultats en pointillés. Gauche : pour l'apprentissage, une base de représentation est apprise via la méthode NMF appliquée aux données audio et visuelles concaténées en vecteurs. Milieu : pour une nouvelle phrase, les données audio sont projetées sur la base apprise. Droite : à partir des coefficients obtenus pour la partie audio, la partie visuelle correspondante est prédite et permet de reconnaitre l'objet nommé.

représentées par des combinaisons linéaires avec des coefficients positifs. Cette méthode permet notamment de découvrir certaines structures sous-jacentes telles que les mots d'un signal sonore ou les parties d'objets (Lee and Seung, 1999). Pour l'apprentissage, nous avons appliqué une version incrémentale de cette méthode (Driesen et al., 2009) à des vecteurs concaténant l'information sonore contenue dans la phrase (à l'aide de coefficients MFCC) et l'information visuelle extraite de l'image associée (Figure 25 et section 3.2.2). Le résultat de cet apprentissage est un ensemble de vecteurs de base ainsi que les coefficients permettant de reconstruire les données d'apprentissage.

Pour la reconnaissance d'une image à partir d'une phrase, seule la partie sonore est utilisée dans un premier temps : elle est projetée sur la base apprise afin de déterminer les coefficients permettant sa reconstruction. A partir de ces coefficients, la partie visuelle correspondante est ensuite générée (Figure 25). Enfin, cette partie visuelle reconstruite est comparée à la description d'images de différents objets pour retrouver l'objet le plus proche qui correspond à la phrase prononcée. Cette méthode permet d'obtenir d'excellents résultats, avec un taux de reconnaissance des objets sur la base ALOI (Geusebroek et al., 2005) de l'ordre de 92% après quelques centaines d'exemples (ten Bosch et al., 2011).

4.1.3 Apprentissage actif et interactif

Au delà des méthodes d'apprentissage particulières, leur mise en oeuvre en robotique dans un cadre interactif avec l'humain est l'occasion de contrôler l'acquisition des données utilisées. Ainsi, comme pour la perception active (section 3.1.5), une méthode d'apprentissage active peut permettre d'améliorer les performances et diminuer

le besoin de données d'apprentissage qui peuvent être coûteuses à acquérir avec un robot.

Nous avons mis en oeuvre une telle méthode pour notre modèle de navigation qualitative (section 2.1.3) qui apprend les modèles de l'environnement en interaction avec un opérateur. Ainsi, le modèle tente de localiser le robot et en cas d'échec va requérir une supervision de l'opérateur pour les données utilisées pour la tentative de localisation. Ces données sont ensuite utilisées pour mettre à jour le modèle.

La figure 6 (section 2.1.3) montre que le taux d'erreur, et donc la fréquence des supervisions demandées à l'utilisateur en utilisant cette stratégie, diminue très rapidement. Pour l'apprentissage d'un modèle complet, cette stratégie permet donc de diminuer le nombre d'exemple d'apprentissage. Par comparaison, (Pronobis et al., 2006) qui utilise des Séparateurs à Vaste Marge obtient des performances de reconnaissance similaires en utilisant 33% des données pour l'apprentissage, alors que nous n'en utilisons que 20%.

Il est également intéressant d'étudier la répartition spatiale des apprentissages. Sans connaissance de la position métrique du robot, notre méthode active concentre naturellement les apprentissages dans les zones ambiguës de l'environnement telles que les passages de portes (figure 6) alors que les zones ouvertes relativement uniformes (les couloirs par exemple) ne sont que peu apprises.

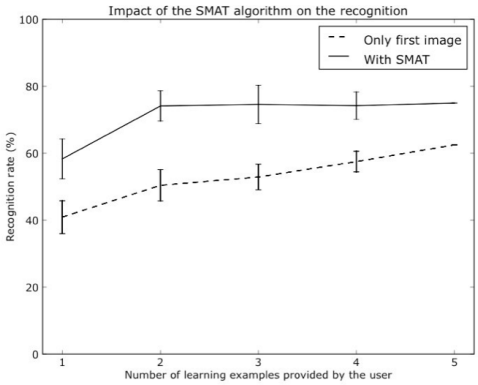

FIGURE 26.: Gain de performances obtenus grâce à l'acquisition autonome de nouveaux exemples d'apprentissage.

Enfin, dans le cadre des travaux de Pierre Rouanet avec l'équipe INRIA FLOWERS, nous avons couplé notre modèle d'apprentissage et de reconnaissance d'objet avec une stratégie d'apprentissage actif (Rouanet et al., 2010). Dans ces travaux, une interface via un téléphone portable permet de désigner à un robot des objets à apprendre. Une fois l'objet désigné, le robot peut tourner autour de cet objet en réalisant un suivi visuel pour obtenir plus d'images d'apprentissage sans avoir besoin de la supervision de l'opérateur. L'ajout de ce mécanisme actif appelé *Active Simultaneous Modelling And Tracking (ASMAT)* permet là aussi d'améliorer sensiblement la qualité de la reconnaissance d'objets (Figure 26).

4.2 APPROCHE DÉVELOPPEMENTALE DE LA ROBOTIQUE

Nous avons souhaité poursuivre les développements dans le domaine de l'apprentissage en robotique selon une approche plus fondamentale, celle de la robotique développementale. Nous avons ainsi participé à une proposition qui a permis d'obtenir un robot iCub, situé à l'ISIR et nous sommes responsable d'un lot de travaux portant sur la perception dans le cadre du projet ANR MACSi (Sigaud et al., 2010). Nous collaborons également sur ce thème avec l'équipe INRIA FLOWERS.

4.2.1 Inspiration biologique

L'objectif technologique principal de la robotique développementale est de concevoir des robots qui sont capables d'apprendre, pendant toute leur existence, de nouveaux savoirs et de nouvelles compétences en interaction avec des utilisateurs qui ne seront pas des spécialistes en robotique. Cette dernière distinction est importante car il ne s'agit pas de permettre à des personnes de programmer de nouveaux comportements ou de nouvelles histoires, mais bien de leur permettre d'enseigner des connaissances au robot comme on le fait pour un enfant.

La robotique développementale (Weng et al., 2001) propose pour cela de s'inspirer de l'enfant et de la façon dont il acquière ses connaissances. Il existe de nombreux travaux, que ce soit en biologie ou en psychologie, portant sur le développement et l'apprentissage chez l'homme, remontant à des précurseurs comme Jean Piaget (Piaget, 1937) ou Vygotsky (Chaiklin, 2003). La robotique développementale s'inspire de ces diverses approches, des études du comportement de l'enfant jusqu'à l'analyse des structures du cerveau impliquées dans l'apprentissage.

Cette inspiration n'est pas à sens unique car la robotique développementale a également pour objectif de permettre de valider des théories de l'apprentissage et du développement. En effet, appliquer une théorie de psychologie développementale à un robot demande en général une réflexion approfondie pour rendre opérationnelle une théorie qui est souvent peu formalisée. Sa mise en oeuvre permet ainsi d'approfondir son étude sur des aspects originaux pour les sciences humaines.

L'idée de concevoir des robots capables d'apprendre n'est pas nouvelle. Ainsi Turing, dès 1950 (Turing, 1950), proposait de tenter de concevoir des robots capables d'apprendre comme des enfants, avec l'espoir qu'il serait plus simple de concevoir ces mécanismes d'apprentissage que de reproduire directement l'intelligence de l'humain adulte. De nombreux travaux d'intelligence artificielle se sont ensuite appliqués à la robotique. Des chercheurs, comme Rodney Brooks dans les années 90 (Brooks, 1991), ont insisté sur l'importance de l'incarnation, de l'interaction avec l'environnement et de l'apprentissage pour développer des robots efficaces. De même, l'approche *Animat* (Meyer, 1995; Guillot and Meyer, 2001) suggère de prendre en compte la biologie et en particulier la façon dont les êtres vivants ont pu résoudre des problèmes pour concevoir des robots s'attaquant à des problèmes similaires.

De très nombreuses recherches ont également été menées sur différentes méthodes d'apprentissage appliquées à la robotique. Cependant, dans la plupart des cas, l'ap-

prentissage demande la création de bases de données d'exemples, ou la préparation de conditions très particulières ; faire apprendre une tâche à un robot demande souvent autant, voir plus de travail au concepteur que la programmation directe. Les méthodes permettant à un robot d'apprendre plusieurs tâches différentes sans une reconfiguration profonde du système, ou en interaction directe, sont rares.

Ainsi, les défis de la robotique développementale demandent de développer de nouveaux algorithmes d'apprentissage qui soient par exemple incrémentaux et stables dans le temps, mais se placent surtout au niveau *système*. Elle s'intéresse, par exemple, au développement des représentations efficaces et évolutives qui permettent d'apprendre de nouvelles tâches sans reprogrammation. Elle développe aussi des méthodes automatiques pour l'acquisition des informations nécessaires à l'apprentissage (Lopes and Oudeyer, 2010). Dans cette optique, la robotique développementale conçoit par exemple des méthodes d'interaction avec l'humain qui permettent au robot d'apprendre mieux et plus rapidement (Rouanet et al., 2010).

4.2.2 Structuration de l'environnement

La capacité à interpréter son environnement et à le segmenter en objets individuels est une des préoccupations de la robotique développementale qui rejoint nos thèmes de recherche. Il existe aujourd'hui de très nombreux algorithmes de traitement d'image, utilisant de plus en plus souvent l'apprentissage, qui permettent de reconnaître des objets (y compris dans des contextes très difficiles, e.g. (Ozuysal et al., 2010)), des visages (Viola and Jones, 2004), ou des lieux dans une ville (Cummins and Newman, 2010). Cependant, ces algorithmes sont tous développés spécifiquement pour leur tâche et lorsque qu'ils utilisent l'apprentissage, ils requièrent en général des bases de données adaptées. A l'inverse, l'homme semble disposer d'une capacité générique pour reconnaître sans efforts tous ces éléments de son environnement.

En particulier, dans toutes les applications robotiques, les éléments à reconnaitre (notamment les objets) sont définis *a priori* par l'humain. Cette approche, utilisable dans de nombreuses applications restera cependant limitée dans sa capacité à découvrir de nouveaux éléments, car il n'y a pas de définition générique du concept d'*objet* qui pourrait permettre de reconnaitre qu'il y a effectivement quelque chose de nouveau. Il est donc intéressant de chercher à créer pour un robot une capacité à créer ce « concept d'objet » (Kraft et al., 2008) de manière non supervisée afin de fournir une base à un éventuel apprentissage à long terme.

Travaux dans le cadre du projet ANR MACSi et de la thèse de Natalia Lyubova

Cette capacité fera naturellement appel aux capacités d'action du robot, qui sont indissociables des capacités de perception. Si l'on suit la théorie des affordances de Gibson (Gibson, 1986), par exemple, un objet est défini par trois caractéristiques :

- il a une taille minimale et maximale par rapport au corps de l'agent
- il a une stabilité temporelle
- il est manipulable par l'agent

La capacité de l'agent à tenter de manipuler un élément de l'environnement est ici cruciale pour permettre de valider que cet élément est bien un objet. Il existe ainsi des approches relativement génériques de l'apprentissage visuel qui permettent à un robot d'apprendre à reconnaitre et manipuler des objets en obtenant les exemples

d'apprentissage de manière autonome, par exemple la théorie des « Object Action Complexes » (Kraft et al., 2008; Krüger et al., 2009).

Dans le cadre du projet MACSi (Sigaud et al., 2010), nous développons actuellement une approche de proto-objets (section 3.1.4) que nous cherchons à rendre suffisamment générique pour apprendre à reconnaitre une grande variété d'éléments de l'environnement. Pour cela, nous étudions différentes représentations de bas niveaux et des méthodes permettant de les combiner en structures locales (section 3.1.4). Ces différentes représentations sont mémorisées incrémentalement dans des dictionnaires hiérarchiques (section 3.1.3) qui jouent le rôle de mémoire à court et long terme. Nous étudions également comment mettre en place de manière incrémentale et rapide des approches de catégorisation non supervisées (section 4.1.2) qui permettront de reconnaitre des éléments visuellement stables et récurrents de l'environnement.

Notre objectif est ensuite de mettre en place un apprentissage des différents éléments de l'environnement de manière ordonnée en s'inspirant des capacités d'apprentissage des enfants. Ainsi, nous envisageons d'apprendre dans un premier temps à reconnaitre des visages qui semblent être parmi les premières choses apprises par les enfants (Butko et al., 2006). Puis d'étendre cet apprentissage aux différentes parties du corps du robot en utilisant la corrélation entre vision et ordres moteurs (Kemp, 2006). Et enfin d'apprendre à reconnaitre des objets en utilisant les capacités de manipulation du robot pour vérifier la présence effective d'objets et permettre l'acquisition autonome des exemples d'apprentissage nécessaires (Kraft et al., 2010).

5 PROJET DE RECHERCHE

Notre projet de recherche s'articule autour de deux axes interdépendants : d'une part l'augmentation de la sémantique pour la cartographie et d'autre part la recherche d'adaptabilité et de robustesse au travers de méthodes d'apprentissage appliquées à la perception. Ces deux axes, menés en parallèle à court terme auront vocation à se rejoindre à plus long terme car ils ont tous les deux pour objectif de fournir une segmentation de l'environnement du robot en éléments de haut-niveau pertinents pour les tâches à réaliser.

5.1 NAVIGATION SÉMANTIQUE

Notre objectif est de développer des méthodes permettant de construire des modèles d'environnement contenant le plus d'information sémantique possible.

5.1.1 Projet PACOM

A court terme, dans le cadre du projet ANR PACOM, nous souhaitons ainsi développer des approches de perception sémantique multi-modales intégrant les données visuelles et de distances, notamment les données obtenues en utilisant des caméras fournissant une image de profondeur qui deviennent depuis peu très accessibles [1]. Nous envisageons pour cela d'intégrer cette information dans l'approche de labelisation sémantique à base de champ de Markov présentée dans la section 3.2.3 et de l'appliquer à la segmentation sémantique d'environnement intérieur.

5.1.2 Cartographie topologique visuelle

Un autre objectif à court terme est de poursuivre le développement de méthodes de cartographie topologique visuelle : d'une part en développant la robustesse et l'applicabilité de ces méthodes en conditions réelles et d'autre part en augmentant également la sémantique des cartes créées, en cherchant par exemple, à obtenir automatiquement des segmentations de l'environnement utiles pour la navigation et l'interaction avec l'homme. En se basant sur une information visuelle panoramique, nous chercherons par exemple à obtenir un découpage automatique en pièces d'un environnement intérieur ou un découpage en routes et intersections d'un environnement urbain.

1. Par exemple la caméra *Kinect* commercialisée par Microsoft

5.1.3 Navigation sémantique multi-modale

Notre objectif à long terme est de développer des approches sémantiques permettant de réaliser des cartes associant des étiquettes hiérarchiques à l'ensemble des éléments de l'environnement (par exemple : pièces, couloir ; mur, porte, fenêtre ; meuble, humain, objets). Ceci suppose des développements sur les méthodes de reconnaissance de ces différents éléments qui pourront bénéficier des nombreux travaux en vision par ordinateur, ainsi que des développements sur des aspects plus spécifiques à la robotique tels que la fusion de données télémétrie/vision et l'exploration active de l'environnement.

Une part importante de ces travaux portera sur les méthodes de représentations de ces connaissances, au travers par exemple d'ontologies (Modayil and Kuipers, 2007). Ces représentations doivent notamment permettre d'utiliser ces informations sémantiques pour le contrôle du robot, ou pour l'amélioration des modèles eux-mêmes, en permettant par exemple de prendre en compte une information de contexte pour la perception.

5.2 APPROCHE DÉVELOPPEMENTALE DE LA PERCEPTION

Nous souhaitons développer en parallèle l'aspect apprentissage appliqué à la perception dans le cadre de l'approche développementale de la robotique.

Les travaux proposés dans la section précédente feront en effet appel à des méthodes d'apprentissage « classiques » dans lesquelles des bases de données d'exemples seront utilisées pour apprendre les éléments à reconnaitre. L'objectif ici est de reproduire dans un robot certaines capacités d'apprentissage des enfants humains en s'inspirant notamment de théories de psychologie développementale. Comme pour l'axe précédent, la finalité est bien de produire des représentations de l'environnement contenant des entités de haut niveau. Toutefois, par rapport à l'utilisation classique de techniques de vision et d'apprentissage en robotique, il s'agit de réaliser cet apprentissage de manière incrémentale, lors de l'interaction avec des humains et sans intervention d'ingénieurs ou de techniciens pour préparer des bases de données ou régler des paramètres.

5.2.1 Projet MACSi

A court terme, dans le cadre du projet ANR MACSi, nous souhaitons développer une première approche de cette idée dans un cadre simplifié. Pour ce projet, la démonstration finale présentera un robot humanoïde iCub, assis à une table avec des objets manipulables posés sur la table et une personne en face de lui.

Dans ce contexte, nous souhaitons faire apprendre au robot à découper son environnement en éléments pertinents. Pour cela, nous utiliserons une approche unique qui permettra de reconnaitre tous ces éléments en séquence selon une trajectoire développementale fixée. Par exemple il serait possible de commencer par la détection de

visage (qui sont les premiers éléments reconnus par les bébés), puis la détection des parties du robot et aller jusqu'à découvrir les objets posés devant lui. Nous développerons donc une approche de représentation visuelle intermédiaire adaptée (section 3.1.4) et intègrerons plusieurs informations de supervision disponibles dans un cadre d'interaction avec l'homme telles que le mouvement, la corrélation de la vision avec les ordres moteurs du robot et le résultat d'actions du robot, ce qui permet par exemple de manipuler un objet pour améliorer sa représentation visuelle.

5.2.2 Apprentissage multi-modal

A plus long terme, nous envisageons l'emploi d'approches plus génériques pour apprendre à segmenter l'environnement en intégrant de nombreuses sources d'information. Nous souhaitons ainsi prolonger les travaux réalisés pour l'apprentissage de mots associés à des objets (section 3.2.2) en y intégrant des informations supplémentaires telle que l'information de distance, ou des informations liées aux actions réalisées par le robot, dans l'esprit des affordances de Gibson.

Cet axe de recherche implique des travaux sur les représentations de l'information perceptuelle, sur la structure de la mémoire, sur des mécanismes d'apprentissage incrémentaux et stables à long terme et sur les méthodes de perception et d'apprentissage actif en interaction avec des humains. Il s'agit donc d'une approche *système* du problème, se rapprochant du développement d'une véritable *architecture cognitive*. Il s'agira donc de définir cette architecture et la trajectoire développementale associée qui permettra d'apprendre à structurer progressivement l'environnement.

Idéalement, cette approche devrait à terme se rapprocher des méthodes de cartographie sémantique en permettant de créer des représentations de haut-niveau de l'ensemble de l'environnement qui soient plus efficacement adaptables au contact de l'homme.

5.3 DOMAINES D'APPLICATION

Ces deux axes de recherche ont pour objectif de créer des modèles riches de l'environnement qui peuvent avoir plusieurs utilisations. Ils peuvent notamment permettre à un opérateur distant d'avoir une bonne compréhension de l'environnement du robot, comme nous l'avons fait dans le projet PACOM, ou au robot d'avoir un comportement plus adapté à son environnement matériel et social.

Nous souhaitons donc poursuivre nos recherches sur les thèmes de la navigation en environnement intérieur ou urbain, appliqué à des robots de service, voir à des véhicules intelligents qui peuvent bénéficier de modèles d'environnement de plus haut niveau sémantique. Nous souhaitons également développer des applications pour des robots en interaction plus forte avec les humains, dans le domaine de la robotique de service et de la robotique d'assistance. Nous envisageons par exemple de travailler sur des problématiques de recherche d'objet dans des environnements quotidiens, en exploitant l'information sémantique pour détecter efficacement ces objets et pour permettre une stratégie de recherche efficace.

BIBLIOGRAPHIE

Agrawal, M. and Konolige, K. (2008). Frameslam : From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5). (Cité page 5.)

Angeli, A. (2008). *Détection visuelle de fermeture de boucle et applications à la localisation et cartographie simultanées*. PhD thesis, Université Pierre et Marie Curie. (Cité page 25.)

Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2008a). A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*. (Cité aux pages i, iii, 13, 14 et 23.)

Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2008b). Incremental vision-based topological slam. In *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*. (Cité aux pages i, iii, 13 et 15.)

Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2009). Visual topological slam and global localization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*. (Cité aux pages i, iii et 13.)

Arel, I., Rose, D. C., and Karnowski, T. P. (2010). Research frontier : deep machine learning–a new frontier in artificial intelligence research. *Comp. Intell. Mag.*, 5 :13–18. (Cité page 25.)

Arleo, A., Smeraldi, F., and Gerstner, W. (2004). Cognitive navigation based on nonuniform gabor space sampling, unsupervised growing networks, and reinforcement learning. *IEEE Transactions on Neural Networks*. (Cité page 6.)

Bay, H., Tuytelaars, T., and Gool, L. (2006). Surf : Speeded up robust features. In *9th European Conf on Computer Vision*. (Cité aux pages 22, 23 et 28.)

Bazeille, S. and Filliat, D. (2010). Combining odometry and visual loop-closure detection for consistent topo-metrical mapping. *RAIRO Operations Research*. (Cité page 15.)

Bazeille, S. and Filliat, D. (2011). Incremental topo-metric slam using vision and robot odometry. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*. (Cité aux pages i, iii et 15.)

Beis, J. S. and Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *In Proc. IEEE Conf. Comp. Vision Patt. Recog*, pages 1000–1006. (Cité page 24.)

Brooks, R. A. (1991). New approaches to robotics. *Science*, 253 :1227–1232. (Cité page 37.)

Brostow, G. J., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57. (Cité page 30.)

Butko, N. J., Fasel, I. R., and Movellan, J. R. (2006). Learning about humans during the first 6 minutes of life. In *Proceedings of the IEEE International Conference on Development and Learning (ICDL)*. (Cité page 39.)

Butko, N. J., Zhang, L., Cottrell, G. W., and Movellan, J. R. (2008). Visual saliency model for robot cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, pages 2398–2403. (Cité page 28.)

Chaiklin, S. (2003). The zone of proximal development in vygotsky's analysis of learning and instruction. In Kozulin, A., Gindis, B., Ageyev, V., and Miller, S., editors, *Vygotsky's educational theory and practice in cultural context*, pages 39–64. Cambridge University Press. (Cité page 37.)

Chatila, R. (2008). Toward cognitive robot companions. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, HRI '08, pages 391–392, New York, NY, USA. ACM. (Cité page 33.)

Chatila, R. and Laumond, J. (1985). Position referencing and consistent world modelling for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-85)*, pages 138–170. (Cité page 5.)

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3) :273–297. (Cité page 34.)

Cummins, M. and Newman, P. (2010). Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*. (Cité page 38.)

Cuperlier, N., Quoy, M., and Gaussier, P. (2007). Neurobiologically inspired mobile robot navigation and planning. *Frontiers in NeuroRobotics*, 1(1). (Cité page 6.)

Dalgalarrondo, A. (2001). *Integration de la fonction perception dans une architecture de contrôle de robot mobile autonome*. PhD thesis, Université Paris-Sud. (Cité page 17.)

Dalgalarrondo, A., Dufourd, D., and Filliat, D. (2004). Controlling the autonomy of a reconnaissance robot. In *SPIE Defense and Security 2004 Symposium. Unmanned Ground Vehicle Technology VI Conference*. (Cité aux pages i, iii et 17.)

Daugman, J. G. (1988). Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics Speech and Signal Processing*, 36(7) :1169–1179. (Cité page 22.)

Davis, R., Shrobe, H. E., and Szolovits, P. (1993). What is a knowledge representation ? *AI Magazine*, 14(1) :17–33. (Cité page 22.)

Driesen, J., ten Bosch, L., and Van hamme, H. (2009). Adaptive non-negative matrix factorization in a computational model of language acquisition. In *Proceedings Interspeech2009 - 10th annual conference of the international speech communication association*. (Cité page 35.)

Duckett, T., Marsland, S., and Shapiro, J. (2000). Learning globally consistent maps by relaxation. In *Proceedings of the International Conference on Robotics and Automation (ICRA'2000)*, pages 3841 – 3846. (Cité page 7.)

Ellis, D. P. W. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab. online web resource. (Cité page 29.)

Everingham, M. (2010). Overview and results of the detection challenge. In *Proceedings of the PASCAL Visual Object Classes Challenge Workshop 2010*. (Cité page 21.)

Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 :36–51. (Cité page 22.)

Filliat, D. (2001). *Cartographie et estimation globale de la position pour un robot mobile autonome (in french)*. PhD thesis, LIP6/AnimatLab, Universite Pierre et Marie Curie, Paris, France. Spécialité Informatique. (Cité aux pages i, iii, 6 et 26.)

Filliat, D. (2007). A visual bag of words method for interactive qualitative localization and mapping. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*. (Cité aux pages i, ii, iii, iv, 10, 23 et 28.)

Filliat, D. (2008). Interactive learning of visual topological navigation. In *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*. (Cité aux pages i, ii, iii, iv, 10 et 27.)

Filliat, D., Benosman, R., and Baillie, J. (2009). Panoramic and active camera for object mapping. Projet ANR CAROTTE. http ://cogrob.ensta.fr/pacom. (Cité aux pages i et iii.)

Filliat, D., Girard, B., Guillot, A., Khamassi, M., Lacheze, L., and Meyer, J.-A. (2004). State of the artificial rat Psikharpax. In Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., and Meyer, J.-A., editors, *From Animals to Animats 8 : Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 3–12, Cambridge, MA. MIT Press. (Cité page 6.)

Filliat, D. and Meyer, J.-A. (2000). Active perception and map-learning for robot navigation. In Meyer, J.-A., Berthoz, A., Floreano, D., Roitblat, H., and Wilson, S. W., editors, *From Animals to Animats 6. Proceedings of the Sixth Conferences on Simulation of Adaptive Behavior.*, pages 246–255. The MIT Press. (Cité aux pages i, ii, iii et 26.)

Filliat, D. and Meyer, J.-A. (2002). Global localization and topological map learning for robot navigation. In Hallam, B., Floreano, D., Hallam, J., Hayes, G., and Meyer, J.-A., editors, *Proceedings of the Seventh International Conference on simulation of adaptive behavior : From Animals to Animats (SAB-2002)*, pages 131–140. The MIT Press. (Cité aux pages i et iii.)

Filliat, D. and Meyer, J.-A. (2003). Map-based navigation in mobile robots - I. a review of localisation strategies. *Journal of Cognitive Systems Research*, 4(4) :243–282. (Cité page 5.)

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *Robotics & Automation Magazine, IEEE*, 4(1) :23–33. (Cité page 17.)

Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J., and González, J. (2005). Multi-hierarchical semantic maps for mobile robotics. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2278–2283. (Cité aux pages 5 et 18.)

Galleguillos, C. and Belongie, S. (2010). Context based object categorization : A critical survey. *Computer Vision and Image Understanding (CVIU)*, 114 :712–722. (Cité page 30.)

Geusebroek, J. M., Burghouts, G. J., and Smeulders, A. W. M. (2005). The amsterdam library of object images. *International Journal of Computer Vision*, 61(1) :103–112. (Cité page 35.)

Gibson, J. J. (1986). *The Ecological Approach To Visual Perception*. Lawrence Erlbaum Associates. (Cité aux pages 21 et 38.)

Giovannangeli, C. and Gaussier, P. (2008). Interactive teaching for vision-based mobile robot : a sensory-motor approach. *IEEE Transactions on Man, Systems and Cybernetics, Part A : Systems and humans*. (Cité page 10.)

Goffettre, R. (2007). Intégration d'information géométrique dans la classification de scènes par sacs de mots. Master's thesis, ENSTA ParisTech. (Cité page 25.)

Gonzalez, F. A. (2010). Semantic image segmentation using markov random fields. Technical report, ENSTA-ParisTech. (Cité page 30.)

Grisetti, G., Stachniss, C., Grzonka, S., and Burgard, W. (2007). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics : Science and Systems*, Atlanta, GA, USA. (Cité page 16.)

Guillot, A. and Meyer, J. (2001). The animat contribution to cognitive systems research. *Journal of Cognitive Systems Research*, 2(2) :157–165. (Cité page 37.)

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151. (Cité page 22.)

Hoyer, P. O. and Dayan, P. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5 :1457–1469. (Cité page 34.)

Jebari, I., Bazeille, S., Battesti, E., Tekaya, H., Klein, M., Tapus, A., Filliat, D., Meyer, C., Ieng, S.-H., Benosman, R., Cizeron, E., Mamanna, J.-C., and Pothier, B. (2011). Multi-sensor semantic mapping and exploration of indoor environments. In *Proceedings of the 3rd International Conference on Technologies for Practical Robot Applications (TePRA)*. (Cité aux pages i, iii, 18, 27 et 28.)

Kemp, C. C. (2006). What can i control ? : The development of visual categories for a robot's body and the world that it influences. In *In 5th IEEE International Conference on Development and Learning (ICDL-06), Special Session on Autonomous Mental Development*. (Cité page 39.)

Kraft, D., Detry, R., Pugeault, N., Baeski, E., Guerin, F., Piater, J., and Krüger, N. (2010). Development of object and grasping knowledge by robot exploration. *IEEE Transactions on Autonomous Mental Development*. (Cité aux pages 22 et 39.)

Kraft, D., Pugeault, N., Baseski, E., Popovic, M., Kragic, D., Kalkan, S., W"org"otter, F., and Kr"uger, N. (2008). Birth of the object : Detection of objectness and extraction of object shape through object action complexes. *International Journal of Humanoid Robotics*, 5(2) :247–265. (Cité aux pages 26, 38 et 39.)

Krüger, N., Piater, J., Wörgötter, F., Geib, C., Petrick, R., Steedman, M., Ude, A., Asfour, T., Kraft, D., Omrcen, D., Hommel, B., Agostino, A., Kragic, D., Eklundh, J., Krüger, V., and Dillmann, R. (2009). A Formal Definition of Object Action Complexes and Examples at Different Levels of the Process Hierarchy. Technical report, EU project PACO-PLUS. (Cité page 39.)

Krüger, N., Pugeault, N., Baseski, E., Jensen, L. B. W., Kalkan, S., Kraft, D., Jessen, J. B., Pilz, F., Nielsen, A. K., Popovi, M., Asfour, T., Piater, J., Kragic, D., and Worgotter, F. (2010). Early cognitive vision as a front-end for cognitive systems. In *Prooceedings of the ECCV Workshop on "Vision for Cognitive Tasks"*. (Cité page 23.)

Krüger, N. and Wörgötter, F. (2004). Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1 :2004. (Cité page 22.)

Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., and Savelli, F. (2004). Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4845–4851, New Orleans, Louisiana. (Cité page 5.)

Kuipers, B. J. and Byun, Y. T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8 :47–63. (Cité page 5.)

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755) :788–791. (Cité page 35.)

Lemaire, T., Berger, C., Jung, I.-K., and Lacroix, S. (2007). Vision-based SLAM : stereo and monocular approaches. *International Journal on Computer Vision*, 74(3) :343–364. (Cité page 13.)

Lopes, M. and Oudeyer, P.-Y. (2010). Guest Editorial Active Learning and Intrinsically Motivated Exploration in Robots : Advances and Challenges. *IEEE Transactions on Autonomous Mental Development*, 2 :65–69. (Cité page 38.)

Lowe, D. (2004). Distinctive image feature from scale-invariant keypoint. *International Journal of Computer Vision*, 60(2) :91–110. (Cité aux pages 22 et 23.)

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press. (Cité page 24.)

Mangin, O., Oudeyer, P.-Y., and Filliat, D. (2010). A bag-of-features framework for incremental learning of speech invariants in unsegmented audio streams. In *Proceedings of the Tenth International Conference on Epigenetic Robotics (EpiRob 2010)*. (Cité aux pages i, iii et 28.)

Meyer, J. A. (1995). The animat approach to cognitive science. In Roitblat, H. and Meyer, J. A., editors, *Comparative Approaches to Cognitive Science*. MIT Press. (Cité page 37.)

Meyer, J.-A. and Filliat, D. (2003). Map-based navigation in mobile robots - II. a review of map-learning and path-planing strategies. *Journal of Cognitive Systems Research*, 4(4) :283–317. (Cité page 5.)

Micusik, B. and Kosecka, J. (2009). Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *IEEE Workshop on Video-Oriented Object and Event Classification (VOEC), held jointly with International Conf. on Computer Vision (ICCV)*. (Cité aux pages 23, 30 et 31.)

Modayil, J. and Kuipers, B. (2007). Autonomous development of a grounded object ontology by a learning robot. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pages 1095–1101. AAAI Press. (Cité page 42.)

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angular sensors. In *Proceedings of the IEEE International Conference On Robotics and Automation (ICRA-85)*, pages 116–121, St. Louis. IEEE Computer Society Press. (Cité page 5.)

Ng, A. Y. and Jordan, M. I. (2001). On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 841–848. (Cité page 34.)

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR06*. (Cité aux pages 16, 23, 24 et 27.)

O'Keefe, J. and Conway, D. (1978). Hippocampal place units in the freely moving rat : Why they fire where they fire. *Experimental Brain Research*, 31 :573–590. (Cité page 6.)

Orabona, F., Metta, G., and Sandini, G. (2008). Attention in cognitive systems. theories and systems from an interdisciplinary viewpoint. chapter A Proto-object Based Visual Attention Model, pages 198–215. Springer-Verlag. (Cité page 26.)

Ozuysal, M., Calonder, M., Lepetit, V., and Fua, P. (2010). Fast Keypoint Recognition using Random Ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (Cité page 38.)

Piaget, J. (1937). *La construction du réel chez l'enfant*. Delachaux et Niestlé. (Cité page 37.)

Pronobis, A., Caputo, B., Jensfelt, P., and Christensen, H. I. (2006). A discriminative approach to robust visual place recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06)*, Beijing, China. (Cité aux pages 11 et 36.)

Pylyshyn, Z. W. (2001). Visual indexes, preconceptual objects, and situated vision. *Cognition*, 80(1-2) :127–158. (Cité page 26.)

Quack, T., Ferrari, V., and Gool, L. V. (2006). Gool. video mining with frequent itemset configurations. In *In Proc. CIVR*, pages 360–369. Springer. (Cité page 25.)

Remazeilles, A. and Chaumette, F. (2007). Image-based robot navigation from an image memory. *Robotics and Autonomous Systems*, 55(4). (Cité page 15.)

Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proc. 9th Int'l. Conf. Computer Vision*, volume 1, pages 10–17. (Cité page 23.)

Rish, I. (2001). An empirical study of the naive Bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, pages 41–46. (Cité page 34.)

Rouanet, P., Oudeyer, P.-Y., and Filliat, D. (2009). An integrated system for teaching new visually grounded words to a robot for non-expert users using a mobile device. In *Proccedings of the 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids09)*. (Cité aux pages i, ii, iii et iv.)

Rouanet, P., Oudeyer, P.-Y., and Filliat, D. (2010). A study of three interfaces allowing non-expert users to teach new visual objects to a robot and their impact on learning efficiency. In *5th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2010*. (Cité aux pages 27, 28, 36 et 38.)

Roy, N. and Thrun, S. (1999). Coastal navigation with mobile robots. In *Advances in Neural Processing Systems 12*, volume 12, pages 1043–1049. (Cité page 26.)

Röfer, T. (2002). Using histogram correlation to create consistent laser scan maps. In *In Proceedings of the IEEE International Conference on Robotics Systems (IROS2002*, pages 625–630. (Cité page 17.)

Sigaud, O., Oudeyer, P., Filliat, D., and Baillie, J. (2010). Modèle pour l'apprentissage du comportement sensorimoteur d'icub. Projet ANR BLANC. http ://-macsi.isir.upmc.fr/. (Cité aux pages ii, iv, 37 et 39.)

Sivic, J. and Zisserman, A. (2003). Video google : A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision (ICCV)*. (Cité aux pages 10, 22 et 23.)

Smith, R., Self, M., and Cheeseman, P. (1988). Estimating uncertain spatial relationships in robotics. In Lemmer, J. F. and Kanal, L. N., editors, *Uncertainty in Artificial Intelligence*, pages 435–461. Elsevier. (Cité page 18.)

Tapus, A., Member, S., and Scassellati, B. (2007). The grand challenges in socially assistive robotics. *IEEE Robotics and Automation Magazine*, 14(1) :1–7. (Cité page 33.)

ten Bosch, L., Oudeyer, P.-Y., and Filliat, D. (2011). Learning by cross-modal association from sub-symbolic continuous data streams. In *En préparation*. (Cité aux pages i, ii, iii, iv, 28, 34 et 35.)

ten Bosch, L., Van Hamme, H., and Boves, L. (2008). Unsupervised detection of words ? questioning the relevance of segmentation. In *Proceedings of the ISCA Tutorial and Research Workshop on Speech Analysis and Processing for Knowledge Discovery*. (Cité page 29.)

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press. (Cité aux pages 5, 6 et 16.)

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55 :189–208. (Cité page 6.)

Turing, A. (1950). Computing machinery and intelligence. *Mind*, (59) :433–460. (Cité page 37.)

Viola, P. A. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2) :137–154. (Cité page 38.)

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*. (Cité page 37.)

Werner, T. (2007). A linear programming approach to max-sum problem : A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(7) :1165–1179. (Cité page 31.)

Yamauchi, B. (1998). Frontier-based exploration using multiple robots. In *Proceedings of the Second international Conference on Autonomous Agents*. (Cité page 19.)

A | CURRICULUM VITAE

David FILLIAT
8, rue Marcel Sembat
94270 Le Kremlin-Bicêtre
Né le 10/09/74 à Clermont-Ferrand (63)
Tel : 01 45 52 54 13
Mail : david.filliat@ensta-paristech.fr
Page web : http://www.ensta-paristech.fr/~filliat/

Expérience professionnelle

Septembre 2008 – présent : Chercheur associé à l'Université Pierre et Maris Curie, Institut des Systèmes Intelligents et de Robotique, équipe SIMA.

Octobre 2005 – présent : Enseignant-Chercheur à l'École Nationale Supérieure de Techniques Avancées – ParisTech, Unité Électronique Informatique. Responsable du thème « Robotique Cognitive »

Janvier 2002 - Octobre 2005 : Ingénieur de l'Armement au Centre Technique d'Arcueil. Chargé d'études et expert en robotique terrestre et aérienne (micro drones).

Formation

1998-2001 : Thèse au Laboratoire d'Informatique de PARIS VI (LIP6) intitulée "Cartographie et estimation globale de la position pour un robot mobile autonome". Sous la direction de Jean-Arcady Meyer. Soutenue le 21 décembre 2001. Mention Très honorable.

1997-1998 : DEA IARFA de l'université PARIS VI. Stage de DEA au laboratoire d'informatique de PARIS VI sur l'évolution de réseaux de neurones pour le contrôle de robot hexapode. Mention Très Bien.

1994-1997 : École Polytechnique. Majeures Mathématiques et Informatique puis Sciences de l'Ingénieur et Calcul Scientifique. Stage de fin d'études au LORIA de Nancy sur un système de navigation pour un robot mobile basé sur des modèles de Markov cachés.

Activité d'enseignement

Responsabilités :

2009 - présent : UMPC : Master SDI, spécialité Systèmes Avancés et Robotique, parcours Robotique Autonome : co-responsable de l'UE Vision pour la robotique.

2006 - présent : ENSTA ParisTech : Co-Responsable de la filière Robotique et Systèmes Embarqués (3ème année).

2006 - présent : ENSTA ParisTech : Responsable de l'enseignement thématique « Intelligence Artificielle » (1ère année).

Enseignements dispensés :

2009 - présent : UMPC, Master SDI – Spécialité SAR, parcours RA : Vision pour la robotique.

2009 - présent : UMPC, Master SDI – Spécialité SAR, parcours SIR : Localisation et Navigation pour la robotique.

2009 - présent : Polytech'Paris-UPMC : Localisation et Cartographie pour la robotique mobile.

2006 - présent : UMPC, Master Informatique – Spécialité IAD : Navigation et Cartographie.

2006 - présent : ENSTA ParisTech, 1ère année, Projet Informatique (C).

2006 - présent : ENSTA ParisTech, 1ère année, Introduction à MATLAB.

2004 - présent : ENSTA ParisTech, 3ème année, Robotique mobile (Localisation, Cartographie, Planification, Evitement d'obstacles).

Support de Cours :

Filliat, D. (2004) : Robotique Mobile, polycopié de cours (162 pages), ENSTA ParisTech.

Activité de recherche

Coordination, animation, coopération :

Depuis 2006 : Responsable du thème Robotique Cognitive, Unité Electronique et Informatique de l'ENSTA ParisTech. Actuellement composé de 2 permanents et de 9 non permanents.

Collaboration avec Pierre-Yves Oudeyer, INRIA Bordeaux sud-ouest. Proposition d'équipe projet INRIA/ENSTA ParisTech FLOWERS sur le thème de la robotique développementale et sociale, création de l'équipe prévue en 2010.

Collaboration en tant que chercheur associé avec l'équipe SIMA de l'ISIR (UPMC), collaboration sur les thèmes de la navigation (Stéphane Doncieux), de la vision appliquée à la robotique (Ryad Benosman).

Membre du Comité Scientifique de Pilotage du GDR Robotique depuis 2010. Participation aux groupes de travail « Méthodologies pour la Robotique » et « Robotique et Enseignement Supérieur » du GDR Robotique.

Membre IEEE, IEEE Technical commitee on Robot Learning, GDR Robotique

Comité de programme European Conference on Mobile Robots (ECMR). Activité de « review » : Journaux : International Journal of Robotics and Automation, IEEE Transactions on Autonomous Mental Development, Robotics and Autonomous Systems Conférences : IEEE International Conference on Robotics and Automation (ICRA), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Advanced Concepts for Intelligent Vision Systems (ACIVS), IEEE International Conference on Development and Learning (ICDL), International Conference On Advanced Robotics. Agences de financement (ICAR) : ANR, Agoranov.

Participation à des projets de recherche :

ANR MACSi (2010-2013) : Responsable de la Tache 1 – Construction d'objets par l'organisation de l'espace perceptuel en proto-objets. Ce projet de robotique

développementale regroupant l'ISIR, l'ENSTA ParisTech, l'INRIA et GOSTAI vise à mettre en place au sein du robot iCub les capacités de perception de l'environnement et de contrôle moteur au travers de méthodes d'apprentissage et en interaction sociale avec des humains.

RAPID Visionav (2009-2011) : Responsable de la Tache 1 – Optimisation algorithmique de la détection de fermeture de boucles. Ce projet regroupant l'ENSTA ParisTech et GOSTAI porte sur l'application des méthodes de détection de fermeture de boucles développées à l'ENSTA ParisTech à la navigation des robots mobiles.

ANR PACOM (2009-2012) : Porteur du projet - Ce projet développe un système pour participer à la compétition d'exploration de bâtiment par des robots dans le cadre de l'appel ANR « Carotte ». Il est conduit par trois partenaires spécialisés dans la navigation pour la robotique mobile (ENSTA ParisTech), la vision panoramique et la vision par ordinateur (ISIR) et l'intégration logicielle pour la robotique mobile (Gostai).

Encadrement de thèses :

Natalia Lyubova (2010- …) : Approche développementale de la perception pour un robot humanoïde.

Cedric Meyer (2010-…) : Théorie de vision dynamique bio-inspirée appliquée à la robotique mobile. Directeur de thèse : Ryad Benosman

Islem Jebari (2009-…) : Cartographie sémantique 3D pour l'exploration d'environnement intérieur. Directeur de thèse : Michel Mauny

Alexandre Chapoulie (2009-…) : Navigation visuelle à grande échelle en milieu urbain. Directeur de thèse : Patrick Rives

Adrien Angeli (2005-2008) : Détection visuelle de fermeture de boucle et applications à la localisation et cartographie simultanées. Directeur de thèse : Jean-Arcady Meyer

Encadrement de post-doctorat :

Stéphane Bazeille (2009-…) : Navigation topologique par apparence visuelle

Encadrement de stages Master et Ecoles d'ingénieur :

Guillaume Duceux (2011) : Evitement d'obstacles 3D pour un robot d'exploration. Stage Master Mécatronique et Microsystèmes (Univ. Franche Conté)

Hassène Tekaya (2010) : Exploration multi objectifs pour un robot mobile. Stage Master SAR (UPMC).

Cédric Meyer (2010) : Détection et localisation d'objets dans des environnements inconnus. Stage Master SAR (UPMC).

Matthias Pointner (2010) : Autonomous navigation and obstacle-avoidance for a wheeled mobile robot. Stage Erasmus Technische Universität München.

Florian Vichot (2008) : Intégration de modules pour un système de navigation topologique. Projet de fin d'étude ENIB.

Bernard Gravier (2008) : Exploration et développement autonome dans le cadre de l'apprentissage par renforcement. Projet de fin d'étude ENSTA ParisTech.

Nicolas Beaufort (2008) : Prototypage d'un système de navigation visuelle pour la robotique mobile. Projet de fin d'étude ESIAL.

Jose Luis Susa Rincon (2007) : Apprentissage supervisé de comportements visuels réactifs. Application au robot Aibo. Projet de fin d'étude ENIB.

Roland Goffettre (2007) : Vision robotique : intégration d'information géométrique dans la classification de scènes par sacs de mots visuels. Projet de Fin d'étude ENSTA ParisTech.

Adrien Angeli (2005) : Cartographie et Localisation simultanées 2D appliquées aux drones. Stage de Master IAD (UPMC).

René Grayden (2005) : Approche sélectionniste pour l'apprentissage par renforcement en grandes dimensions. Stage de Master IAD (UPMC).

Jury de thèses :

Ayman Zureiki (16/09/08, Examinateur) : Fusion de données multi-capteurs pour la construction incrémentale du modèle tridimensionnel texturé d'un environnement intérieur par un robot mobile

Thomas Degris (26/04/07, Examinateur) : Apprentissage par renforcement dans les processus de décision Markoviens factorisés

Christopher Mei (09/02/07, Examinateur) : Couplage Vision Omnidirectionnelle et Télémétrie Laser pour la Navigation en Robotique

Publications scientifiques

Revues internationales à comité de lecture :

1. Angeli, A., Filliat, D., Doncieux, S., & Meyer, J.-A. (2008) *A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words*. IEEE Transactions On Robotics, Special Issue on Visual SLAM.
2. Girard, B., Filliat, D., Meyer, J., Berthoz, A., and Guillot, A. (2005) *Integration of navigation and action selection functionalities in a computational model of cortico-basal ganglia-thalamo-cortical loops.* Adaptive Behavior, 13(2):115-130.
3. Filliat D. and Meyer, J.A. (2003*). Map-based navigation in mobile robots - I. A review of localization strategies*. Journal of Cognitive Systems Research. 4, 4, 243-282.
4. Meyer, J.A. and Filliat D.(2003). *Map-based navigation in mobile robots - II. A review of map-learning and path-planning strategies*. Journal of Cognitive Systems Research. 4, 4, 283-317.
5. Filliat, D. , Kodjabachian, J. and Meyer, J.A. (1999). Evolution of neural controllers for locomotion and obstacle-avoidance in a 6-legged robot. Connection Science. vol. 11. pp. 223--240. 1999

Chapitres d'ouvrages :

6. Meyer, J.-A., Doncieux, S., Filliat, D., & Guillot, A.: Evolutionary Approaches to Neural Control of Rolling, Walking, Swimming and Flying Animats or Robots. Biologically Inspired Robot Behavior Engineering. Editted by Duro, R.J. and Santos, J. and Grana, M.. Springer-Verlag. pp. 1--43. 2002

Conférences Internationales :

7. *Bazeille, S. & Filliat, D. (2011) Incremental topo-metric SLAM using vision and robot odometry. Proceedings of the International Conference on Robotics and Automation (ICRA).*
8. *Jebari, Bazeille, Battesti, Tekaya, Klein, Tapus, Filliat, Meyer, Ieng, Benosman, Cizeron, Mamanna, & Pothier (2011) Multi-sensor semantic mapping and exploration*

of indoor environments. Proceedings of the 3rd International Conference on Technologies for Practical Robot Applications (TePRA).

9. Mangin, O., Filliat, D. and Oupeyer P.-Y. (2010): *A bag-of-features framework for incremental learning of speech invariants in unsegmented audio streams.* In proceedings of the tenth international conference on epigenetic robotics: Modeling Cognitive Development in Robotic Systems (Epirob 2010).
10. Angeli, A., Filliat, D., Doncieux, S., & Meyer, J.-A. (2009) *Visual topological SLAM and global localization.* Proceedings of the International Conference on Robotics and Automation (ICRA09)
11. Bazeille, S. and Filliat, D. (2009) *Combining Odometry and Visual Loop-Closure Detection for Consistent Topo-Metrical Mapping*. Proccedings of the conference on COGnitive systems with Interactive Sensors (COGIS2009).
12. Rouanet, P., Oudeyer, P.-Y. & Filliat, D. (2009) *An integrated system for teaching new visually grounded words to a robot for non-expert users using a mobile device.* Proceedings of the International conference on humanoid robotics (Humanoïds09).
13. Angeli, A., Filliat, D., Doncieux, S., & Meyer, J.-A. (2008) *Incremental vision-based topological SLAM*. Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008).
14. Angeli, A., Filliat, D., Doncieux, S., & Meyer, J.-A.(2008) *Real-Time Visual Loop-Closure Detection.* Proceedings of the International Conference on Robotics and Automation (ICRA 2008).
15. Filliat, D. (2008) *Interactive learning of visual topological navigation*. Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008).
16. Filliat, D. (2007) *A visual bag of words method for interactive qualitative localization and mapping.* Proceedings of the International Conference on Robotics and Automation (ICRA 2007).
17. Angeli, A., Filliat, D., Doncieux, S., & Meyer, J.-A. (2006) *2D Simultaneous Localization And Mapping for Micro Aerial Vehicles*. European Micro Aerial Vehicles (EMAV 2006).
18. Dalgalarrondo, A., Dufourd, D. and Filliat, D. (2004) *Controlling the autonomy of a reconnaissance robot.* SPIE Defense & Security 2004 Symposium. Unmanned Ground Vehicle Technology VI Conference, Orlando (FL), USA, 12-16 april 2004.
19. Filliat, D., Girard, B., Guillot,A., Khamassi, M., Lacheze, L., and Meyer, J.A. (2004) *State of the artificial rat Psikharpax*. From Animals to Animats 8: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior. Cambridge, MA. pp. 3--12.
20. Girard, B., Filliat, D., Meyer, J.-A., Berthoz, A., and Guillot, A. (2004) *An integration of two control architectures of action selection and navigation inspired by neural circuits in the vertebrates: The Basal ganglia*. Connectionist Models of Cognition and Perception II, Proceedings of the Eighth Neural Computation and Psychology Workshop. pp. 72—81.
21. Filliat, D. and Meyer, J.A. (2002) *Global localization and topological map learning for robot navigation*. Proceedings of the Seventh International Conference on simulation of adaptive behaviour: From Animals to Animats (SAB-2002). pp. 131--140.
22. Filliat, D. and Meyer, J.A. (2000) *Active perception and map-learning for robot navigation*. From Animals to Animats. Proceedings of the Sixth Conferences on Simulation of Adaptive Behavior.. pp. 246--255.
23. Filliat, D., Kodjabachian, J., and Meyer, J.A. (1999) *Incremental evolution of neural controllers for navigation in a 6-legged robot.* Proceedings of the Fourth International Symposium on Artificial Life and Robotics. pp. 745--750.

B SÉLECTION D'ARTICLES

Nous présentons dans cette annexe les principaux articles décrivant nos travaux. Ces articles sont présentés dans l'ordre dans lequel ils sont cités dans le manuscrit :

1. Filliat, D. and Meyer, J.-A. (2002). Global localization and topological map learning for robot navigation. In Hallam, B., Floreano, D., Hallam, J., Hayes, G., and Meyer, J.-A., editors, *Proceedings of the Seventh International Conference on simulation of adaptive behavior : From Animals to Animats (SAB-2002)*, pages 131–140. The MIT Press. (page 61)
2. Filliat, D. (2008). Interactive learning of visual topological navigation. In *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*. (page 71)
3. Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2008a). A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*. (page 79)
4. Bazeille, S. and Filliat, D. (2011). Incremental topo-metric slam using vision and robot odometry. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*. (page 91)
5. Dalgalarrondo, A., Dufourd, D., and Filliat, D. (2004). Controlling the autonomy of a reconnaissance robot. In *SPIE Defense and Security 2004 Symposium. Unmanned Ground Vehicle Technology VI Conference*. (page 99)
6. Jebari, I., Bazeille, S., Battesti, E., Tekaya, H., Klein, M., Tapus, A., Filliat, D., Meyer, C., Ieng, S.-H., Benosman, R., Cizeron, E., Mamanna, J.-C., and Pothier, B. (2011). Multi-sensor semantic mapping and exploration of indoor environments. In *Proceedings of the 3rd International Conference on Technologies for Practical Robot Applications (TePRA)*. (page 111)
7. Rouanet, P., Oudeyer, P.-Y., and Filliat, D. (2009). An integrated system for teaching new visually grounded words to a robot for non-expert users using a mobile device. In *Proccedings of the 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids09)*. (page 117)
8. Mangin, O., Oudeyer, P.-Y., and Filliat, D. (2010). A bag-of-features framework for incremental learning of speech invariants in unsegmented audio streams. In *Proceedings of the Tenth International Conference on Epigenetic Robotics (EpiRob 2010)*. (page 125)

Global localization and topological map-learning for robot navigation

David Filliat*
*DGA/Centre Technique d'Arcueil
16 bis, av. Prieur de la Côte d'Or
94114 Arcueil Cedex - France
david.filliat@etca.fr

Jean-Arcady Meyer**
**AnimatLab-LIP6
8, rue du capitaine Scott
75015 Paris - France
jean-arcady.meyer@lip6.fr

Abstract

This paper describes a navigation system implemented on a real mobile robot. Using simple sonar and visual sensors, it makes possible the autonomous construction of a dense topological map representing the environment. At any time during the mapping process, this system is able to globally localize the robot, i.e. to estimate the robot's position even if the robot is passively moved from one place to another within the mapped area. This is achieved using algorithms inspired by Hidden Markov Models adapted to the on-line building of the map. Advantages and drawbacks of the system are discussed, along with its potential implications for the understanding of biological navigation systems.

1 Introduction

The word navigation refers to all the strategies that may be used by a robot to purposely move in its environment. Such strategies range from simple visible goal heading behavior to complex map-based navigation that allows the planification of movements to arbitrary distant goals (Trullier et al., 1997). Using the latter strategies basically raises three sub-problems : *map-learning*, which concerns the construction of a map representing the environment, *localization*, which concerns the estimation of the robot's position inside this map and *planification*, which concerns the design of a plan to reach a given goal.

Every navigation strategy may call upon two sources of information. The first is the idiothetic source that provides information about the robot's movements using internal sensors such as accelerometers. This information can be directly expressed in a metrical space. The second one is the allothetic source that provides information about the robot's position inside its environment using external sensors such as sonar sensors or a camera. The characteristics of these two sources are complementary : while idiothetic information suffers from cumulative errors that make it unreliable for long-term position estimation, allothetic information suffers from the *perceptual aliasing* problem that prevents the robot from distinguishing between two places. Therefore, the efficiency of a navigation system usually relies on its capacity to efficiently combine these two types of information.

It is important to note that allothetic information can be used in two different ways. The first makes use of a metrical model of the sensors, which permits the allothetic data to be expressed in the metrical space of idiothetic information. This is, for example, the case for sonar data used to estimate the position of obstacles in a metrical map of the environment (Moravec and Elfes, 1985). The second way avoids any use of metrical models of the sensors and directly resorts to allothetic information to compare and recognize different positions. This is, for example, the case when the colors of the environment are used to recognize a position in a topological map (Ulrich and Nourbakhsh, 2000). This paper will limit itself to methods that use allothetic sensors without any associated metrical model. Indeed, this choice makes a much more general use of allothetic data possible, as it does not require sensors measuring metrical properties of the environment. This way, information like a color, an odor or a temperature can be used to map the environment. Moreover, such simple use of allothetic data seems more representative of the way an animal like a rat builds an internal model of its environment.

Without metrical models of the sensors, however, a navigation system will have to cope with some limitations. Most of these limitations stem from the fact that it is impossible to infer what should be perceived at a distant position without actually going there. For example, it is easy, with a metrical sensor model, to infer that a wall perceived two meters away will be perceived as being one meter away if the robot moves one meter in the direction of this wall. On the contrary, such an inference is impossible without using a metrical sensor model. Consequently, a map-learning system will only provide information about positions that have already been visited at least once. As will be shown in the re-

mainder of this paper, this limitation must be dealt with by the map-learning and localization procedures.

The main issue with map-based strategies lies in the necessity of simultaneously tackling localization and map-learning problems. The difficulty arises from the *chicken and egg* status of these problems (Yamauchi et al., 1999). In other words, a map is necessary to estimate the position, while knowing the position is necessary to update the map. It is true that the localization problem when a map is given a priori has been given efficient solutions (Thrun et al., 1999). Notably, some models are able to tackle the *lost robot problem*, i.e. the estimation of the robot's position without any initial cues about its position. Unfortunately, the corresponding models that are able to *globally localize* a robot are difficult to extend to on-line map-learning.

In the context of animat research, strong emphasis is placed on autonomy. A map-based navigation system should therefore make it possible to accurately localize an animat in any, eventually initially unknown, environment without human intervention. These requirements are met by global localization models that build environmental maps on-line. The model described in this paper affords solutions to such requirements. Moreover, for the reasons stated above, this model does not make use of any metrical sensor model. It draws inspiration from the literature on bio-mimetic navigation systems, on the one hand, and from purely robotic navigation systems, on the other hand. Several improvements to the simulation model presented by Filliat and Meyer (2000) will be described here, together with new results that were obtained with a real robot implementation.

2 Global localization and map-learning

Localization models described in the literature basically pertain to three categories called respectively *direct-position inference*, *single-hypothesis tracking* and *multiple-hypothesis tracking* (Filliat and Meyer, 2002).

2.1 Direct-position inference

These models (e.g., Franz et al., 1998, Gaussier et al., 2000) call upon environments and sensory capacities that are not subject to perceptual aliasing. Allothetic information is supposed to directly provide an unambiguous estimate of the position, without the need to use any idiothetic information. These models therefore heavily rely on perceptual systems that are able to discriminate between a great number of positions. However, such an hypothesis about the absence of perceptual aliasing within a whole environment is hard to assume *a priori* in any initially unknown environment.

2.2 Single-hypothesis tracking

These models (e.g., Smith et al., 1988, Dedeoglu et al., 1999) take the perceptual aliasing issue into account and solve it by using idiothetic information to disambiguate positions. This information is used to estimate the current position relative to the previous one, and this estimate is used to limit the search space of the position that corresponds to current allothetic data. Assuming that the restrained search area no longer exhibits perceptual aliasing, the corresponding position is unique. This mechanism allows a single position hypothesis to be tracked, as the alternative positions that would correspond to the same allothetic data are simply discarded.

This method is *local* in the sense that the current position is searched for only in the vicinity of the previous position estimate and not over the whole map. As a consequence, an initial position estimate has to be provided to the system either by a separate direct position inference mechanism or by an operator. This requirement limits the robot's autonomy and moreover precludes future correct position estimation if the current estimate should accidentally prove false.

2.3 Multiple-hypothesis tracking

A solution to avoid the dependence on an initial position estimate in perceptually aliased environments is to track multiple hypotheses of the robot's position. According to this scheme, instead of discarding the positions corresponding to current allothetic data that do not match the previous position estimate, these positions are memorized as alternative hypotheses of the robot's position. All these hypotheses are subsequently tracked in parallel and their relative credibilities are monitored. At every moment, the most credible hypothesis is considered as the robot's current position.

This approach allows a *global localization* that is not tied to an initial position estimate. Moreover, the set of concurrent hypotheses may be empty and may be initialized with all the positions that correspond to the first allothetic information gathered in the environment. Therefore, this approach solves the *lost robot problem*, and it affords a high degree of autonomy to the localization process.

The corresponding implementation may call upon the explicit process of monitoring several possible positions in parallel (Piasecki, 1995), or it may call upon Partially Observable Markov Decision Processes (Simmons and Koenig, 1995, Fox et al., 1998). These latter solutions may be viewed as implicit multiple-hypothesis tracking, where each possible position in the map is considered as a position hypothesis. This solution already yielded highly successful robots operating in challenging environments (Thrun et al., 1999).

2.4 Map-learning

From a recent review of map-learning strategies in robots (Meyer and Filliat, 2002), it appears that combining map-learning with direct position inference is relatively straightforward as it simply entails adding to the map allothetic situations that have never been seen before.

A lot of models also combine map-learning with single-position tracking methods (Arleo and Gerstner, 2000, Dedeoglu et al., 1999) because this approach still works when the robot gets outside the area already mapped. Indeed, in such case, it is straightforward to insert a new position in the map, because it is defined relatively to a previously known position.

On the contrary, combining map-learning with multiple-hypothesis tracking algorithms is more difficult. The reason is that these algorithms rely heavily on the completeness of the map to estimate the relative credibilities of the different position hypotheses. This estimation entails comparing what the robot currently perceives with what it should perceive in each of the possible positions monitored. Therefore, when the map is incomplete - which is the case during map-learning - this estimation is difficult, as the robot may be either inside or outside the currently mapped area. If it is inside, the global localization procedure can estimate the robot's position; if it is not, this procedure cannot be used.

Various attempts have been made to overcome this difficulty while nevertheless combining global localization with map-learning. A first method is to use off-line mapping algorithms that build a map corresponding, with the highest possible probability, to a set of data gathered by the robot (Shatkay and Kaelbling, 1997). However, this method does not meet our requirement of autonomy because localization and map-learning are to be separated.

A second method that works on-line is to use powerful distance sensors, along with associated metrical models, in order to prevent the robot from traveling outside the mapped area (Thrun et al., 2000). Indeed, as argued in the introduction, metrical sensor models make it possible to build a map that extends beyond the current robot's position. Accordingly, frequently estimating the robot's position guarantees that it always remains within the mapped area.

A third method will be used here, which combines global localization and map-learning without resorting to any metrical sensor model. This method entails frequently checking whether the robot is in the mapped area or not. If such is the case, a global localization algorithm can be used directly. If not, a single hypothesis tracking method based on the previous positions is used temporarily, until the robot re-enters the mapped area. To decide between these two alternatives, Filliat and Meyer (2000) proposed to simply use the credibility of the most credible among the concurrent position hypotheses. Should this credibility fall below a given threshold, the robot would be considered to be outside the mapped area. However, additional experiments with such a procedure showed it to be brittle, because the corresponding threshold needed to be changed according to the particular environment mapped. Moreover, large uncertainties in the robot's position, which lead to low credibilities of the concurrent hypotheses, always led to believe that the robot was outside the mapped area, thus rendering the mapping process quite unstable.

This paper describes an updated model where the decision between the two cases calls upon an heuristic based on the variation of the sum of credibilities of the various hypotheses. This heuristic, that will be described later on, efficiently detects when the robot exits the mapped area, thus affording the model a substantial gain in robustness, notably because the corresponding parameters become independent of the environment.

3 The-model

This section outlines a simplified version of the model that assumes that panoramic sensors are used. Experimental results presented further were obtained with directional sensors and active perception strategies described in Filliat and Meyer (2000) and Filliat (2001).

3.1 Structure

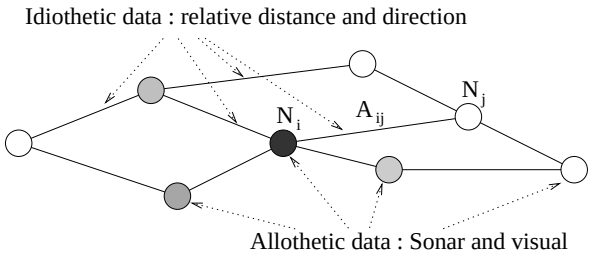

Figure 1: The topological map used in the model.

The map built by the system is a dense topological map, the nodes of which represent close positions in the environment (with a mean spacing of 25 cm). Each node stores the allothetic data that the robot can perceive at the corresponding place in the environment. A link between two nodes memorizes at which distance and in which direction the corresponding places are positioned relatively to each other, as measured by the robot's idiothetic sensors (Figure 1). All the directions used in the model are absolute directions, assuming a fixed reference direction given by a magnetic compass. The robot's position is represented by an activity distribution over the nodes : activity A_i of node i represents the probability that the robot is at the corresponding position. These probabilities are estimated using allothetic and idiothetic

data gathered by the robot, as will be described in section 3.3.

The model iterates the following steps that are explained in the paragraphs below :

- Update the activity of each node in the map;
- Recognize a node as corresponding to the robot's current position or create a new one;
- Update visual and sonar data stored in the recognized node using the current allothetic data;
- Update the idiothetic data stored in the links;
- Choose the direction of the next move in order to explore the environment or to reach a goal.

3.2 Model inputs

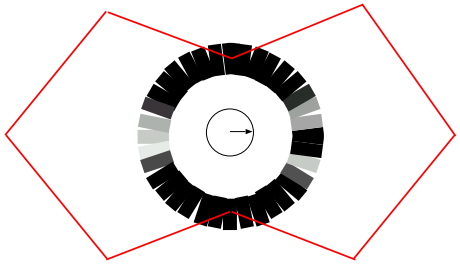

Figure 2: Schematics of allothetic data used in the model. The broken line joins the points detected by sonar sensors in eight absolute directions. The rectangles arranged on a circle indicate the mean grey-level perceived in the corresponding direction by the camera.

Two series of allothetic data are used in the model : sonar data and visual data (Figure 2). Sonar data are gathered through a 16-sonar belt and aggregated into eight virtual sensors that provide distances to obstacles in eight absolute directions. Visual data are gathered by an omnidirectional camera and down-sampled to the values of 36 virtual sensors that measure the mean grey-level of the environment in 36 absolute directions.

Both sonar and visual allothetic data are associated with a procedure P_O that compares two perceptions O_M and O_P. This procedure, which returns 1 if the two perceptions are identical, and decreases to 0 more quickly the more the perceptions are different, is used to estimate the probability that the robot is at a position characterized by data O_M, given the currently perceived allothetic data O_P. In the experiments described below, we used the following function[1] :

$$P_O(O_M/O_P) = \sqrt[l]{\prod_{k=1}^{l} F(O_M^k - O_P^k)}$$

[1]This procedure is adapted to the case of partial data when a directional camera is used. See Filliat (2001) for details.

where O_M^k and O_P^k are the values of allothetic data in the absolute direction k, l is the total number of directions for the considered sensor - i.e., eight for sonar data and 36 for visual data - and F is a Gaussian function given by $F(x) = e^{-x^2/K^2}$. The parameter K is chosen empirically for each sensor so as to give $P_O = 10^{-6}$ for maximally different sensor values. The model seems robust with respect to this parameter, since the same value was efficiently used for all simulated and real experiments.

Idiothetic data are used to estimate the probability that the robot has moved from one node in the map to another. Given a displacement of direction θ_{od} and length r_{od} measured by the robot's odometry, the probability of having moved from node A to node B is :

$$P_D(AB/od) = E_1 \times E_2$$

with :

$$E_1 = exp\left(\frac{-(\theta_{od} - \theta_{AB})^2}{L^2}\right)$$

$$E_2 = exp\left(\frac{-(r_{od} - r_{AB})^2}{M^2}\right)$$

where θ_{AB} and r_{AB} are the direction and length of the link between nodes A and B, L and M are empirically set to $L = 30$ *degrees* and $M = 20$ *cm* through statistics gathered on the moves interspersed with activity updates. Here also, the same values have been used in all simulated and real experiments.

3.3 Activity updates

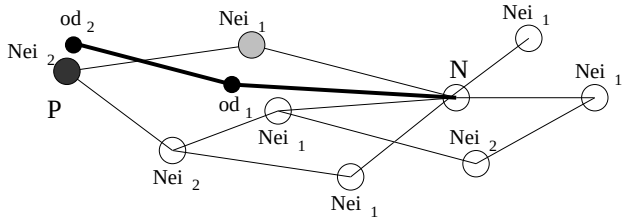

Figure 3: Illustration of the use of idiothetic data for activity updates. Nei_k is the set of all the nodes linked to node N by k connections, and od_k is the position of the robot at time $t-k$ as measured by the odometry relatively to node N. In this example, the activity of node N will be a function of the activity of node P, at time $t-2$ (see text for details).

The activity of each node is updated each time the robot has moved by a given distance (50 cm in the experiments). Such updates are directly inspired by the equation used in POMDP-based navigation models (Simmons and Koenig, 1995) and are adapted to the irregular structure of our model. Idiothetic data are first integrated using the equation :

$$A_i(t) = \max_{k\in[1..K]} \left(\max_{j\in Nei_k(i)} (A_j(t-k) \times P_D(ij/od_k)) \right)$$

where $Nei_k(i)$ is the set of all the nodes linked to node i by k connections, $A_j(t-k)$ is the activity of node j at time $t-k$, and od_k is the position of the robot at time $t-k$ as measured by the odometry relatively to node i.

The effect of this equation is to estimate the probability of the robot's being at node i, taking into account the node j that best fits the robot's path over K past time-steps (see Figure 3). The sum S_a of the activities of all the nodes is then calculated. It will be used to decide whether the robot is in the mapped area or not (see next section).

Then, allothetic data O_P are integrated using :

$$A_i(t+1) = A_i(t) \times P_O(O_i, O_P)$$

The effect of this equation is to increase the activities of nodes characterized by allothetic data that match the current perceptions, and to decrease the activities of the other nodes. Activities are then normalized such that their sum equals 1.

3.4 Position estimation

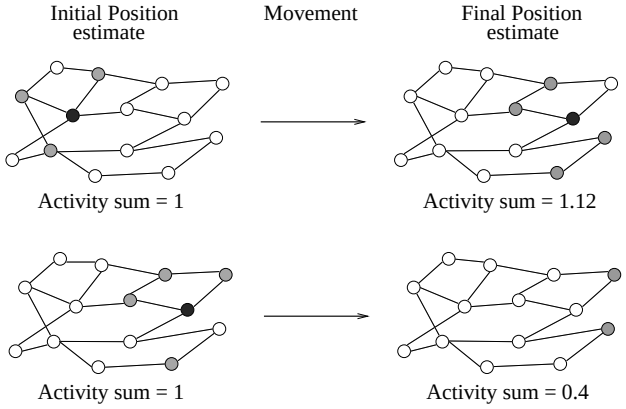

Figure 4: Illustration of the heuristic used to detect whether the current position is in the mapped area or not. When the robot is in the mapped area, the sum of the activities remains approximately constant (top half of the figure) while, if the robot exits the mapped area, the sum decreases (bottom half of the figure).

The model presented so far estimates the robot's most probable position, assuming that this position is part of the map. However, during map-learning, the robot can get out of the mapped area. To decide if the robot has exited the mapped area, an heuristic based on the variation of the sum of the activities before and after the integration of idiothetic cues is used. The idea underlying this heuristic is that, when the robot exits the mapped area, the sum S_a of the activities should suddenly decrease (see Figure 4). If the robot remains in the mapped area, on the contrary, this sum should either increase or remain stable.

Taking into account that the sum of activity is 1 before idiothetic cue integration, the algorithm used to estimate the position is then :

- If $S_a \geq 1$, the node with the highest activity is recognized as the current position.
- If $S_a < 1$, the robot's position is estimated using odometry information gathered since the last recognized node. If this position falls close to an existing node, this node is recognized; otherwise, a new node is added to the map[2]. Such a procedure amounts to temporarily using a single-position tracking method.

3.5 Map updates

Once the node corresponding to the current position has been determined, the allothetic data that characterize it are updated using the newly perceived data.

The direction and distance that correspond to the link between the previously recognized node and the current one are also updated using the newly measured displacement. To achieve map consistency, the values of all the links in the map are then updated using the relaxation algorithm of Duckett et al. (2000). In this context, a map is considered to be consistent if, when two different paths link two nodes, the relative positions of these nodes, calculated by summing the connection data along these two paths, are identical. Basically, the relaxation algorithm "shakes" the relative positions of all the nodes in the map so as to make these relative positions as close as possible to their measured values, thereby resulting in a globally coherent map.

3.6 Exploration strategy

Once the map has been updated, the exploration of the environment resumes. The exploration strategy used in the model aims at limiting localization errors and at ensuring exhaustive exploration. As global localization is efficient only when the robot is in the mapped area, the exploration strategy limits the distance that the robot may travel in an unmapped area. This is implemented thanks to a mechanism that retraces the recent route backwards if the model consecutively creates five nodes, i.e. if the heuristic mentioned above detects that the robot is outside the mapped area during five consecutive

[2]It should be noted that the heuristic thus used has a tendency to over-estimate the novelty of a position, which results in having any unmapped position always being correctly recognized as new. However, it also often causes a position previously mapped to be classified as new. This over-estimation is compensated for by verifying the existence of a node close to the position estimated before creating a new one.

time-steps. When this mechanism is not active, on the contrary, the direction of movement is chosen towards the less explored area, i.e. the direction free of obstacles where there are fewer nodes in the map, so as to ensure exhaustive exploration.

3.7 Path planning

If a goal is assigned to the robot, a movement is planned towards this goal. To achieve this, a *policy*, determining in which direction D_i to move from each node i of the map to reach the goal, is calculated using a simple spreading-activation algorithm starting from the goal. The direction of the next move is then chosen according to a voting method (Cassandra et al., 1996). A score is accordingly calculated for 36 sectors of 10 degrees surrounding the robot. This score is the sum of the activities of the nodes whose associated direction falls in this sector :

$$V(d) = \sum_{d-5<D_i<d+5} A_i$$

where $V(d)$ is the score of the sector of direction d, D_i is the direction of the goal associated with node i, and A_i is the activity of node i. The direction to be taken by the robot corresponds to the sector that achieves the highest score.

A detour mechanism may also be triggered when the planned trajectory to the goal turns out to be blocked by an unforeseen obstacle (Tolman, 1948). In such a case, the contradiction between planned movements that would lead the robot to cross the obstacle and the local obstacle-avoidance procedures that repel the robot from this obstacle generates an oscillatory behavior in front of the obstacle. These oscillations are detected by a continuous check of the robot's progression and a threshold is used to detect when too low a progression indicates it is probably impossible to reach the goal. The nodes that are close to the robot's position are then excluded from the planning process, which is entirely repeated. This results in a new policy that avoids the blocked position and leads the robot to the goal by a different route whenever possible (Filliat, 2001).

4 Experimental results

The model has been implemented on a Pioneer 2 mobile robot (see Figure 5). This robot is equipped with 16 sonar sensors and a directional camera. Although a magnetic compass could be used to estimate the absolute direction, this sensor turned out to be inefficient in our environment because of numerous magnetic disturbances. In the current system, the direction is therefore estimated using the robot's odometry, and its error is periodically compensated for by manually aligning the robot with a reference direction. This correction is made every 50 time-steps, i.e. approximately every 10 minutes. A set of low-level procedures allows local obstacle-avoidance during navigation.

Figure 5 shows a map obtained by the system in the corridors of our laboratory, this map being superimposed on an architectural sketch of the environment. It was created in 2000 time-steps in approximately six hours of operation, most of this time being consumed in stopping and starting the robot and in orienting the camera at each time-step. This time could be significantly reduced by the use of an omnidirectional camera that would allow the system to operate without stopping the robot at each time-step. Be that as it may, the map thus obtained correctly reproduces the structure of the laboratory and permits the robot's position to be estimated precisely. Figure 6 shows part of the robot's trajectory, as estimated either by the whole localization system or by a sub-part of this system that called upon the robot's odometry only. The trajectory estimated by the whole system is closer to the real trajectory, because it remains in the open area and does not cross any wall, thus demonstrating that the localization system is efficient.

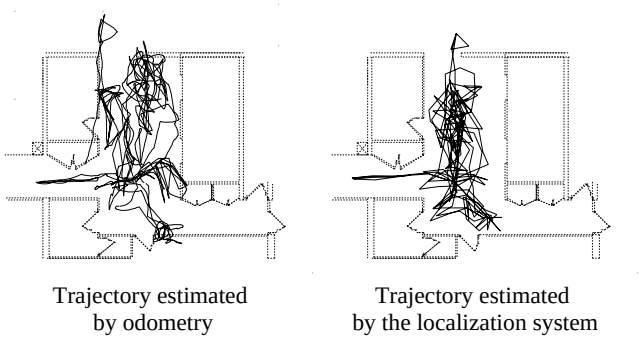

Figure 6: Comparison of two procedures to estimate the robot's trajectory. Left: results obtained with odometry alone. Right: results obtained with the full navigation system.

Moreover, the localization algorithm effectively achieves global localization most of the time. Indeed, it frequently computes the robot's position using node activities instead of using the position-tracking method that is temporarily triggered when the navigation system detects that the robot is outside the mapped area (Figure 7).

We carried out specific experiments to demonstrate this global localization capacity. In particular, we stopped the localization system when the robot was correctly localized at position A and subsequently manually moved it to position B in the environment of Figure 5. The standard localization and exploration process were then resumed without providing the system any cue about this displacement. Figure 8 shows the error in the estimation of the position during the subsequent lo-

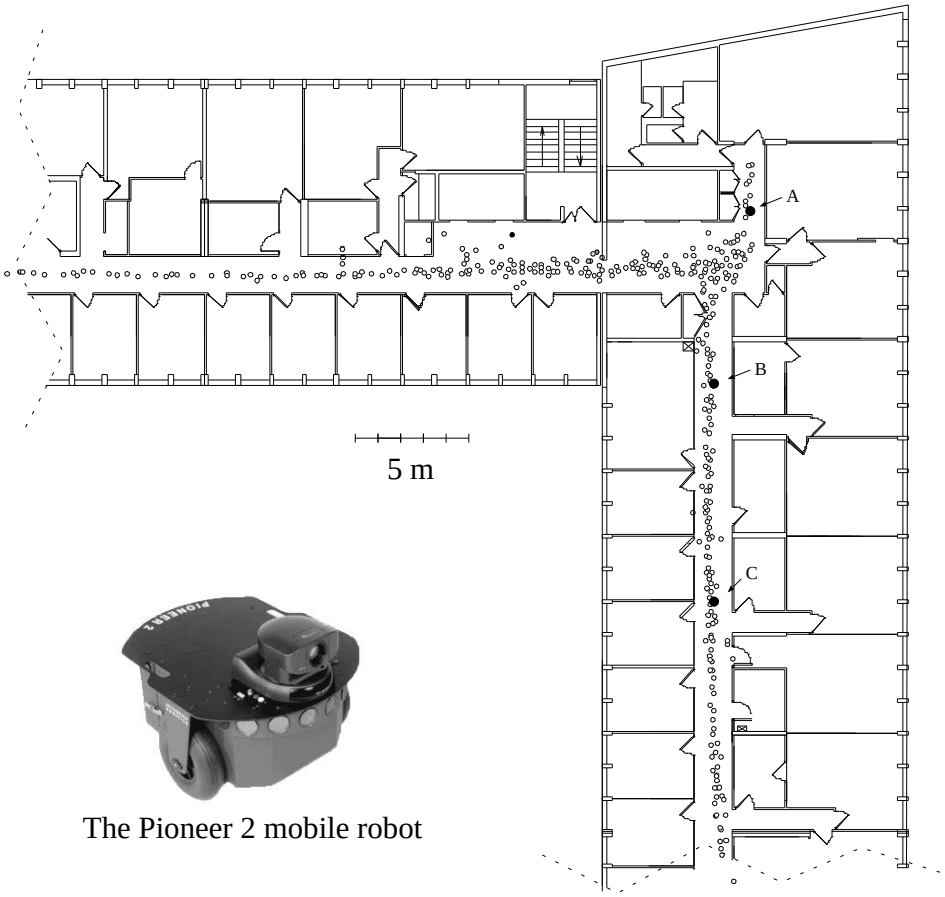

Figure 5: An example of a map created in the corridors of our laboratory. The map is superimposed on an architectural sketch of the environment.

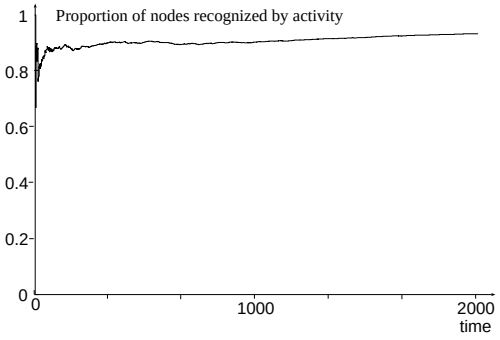

Figure 7: Proportion of the number of nodes that are recognized by the global localization system using node activities relatively to the total number of recognized nodes.

calizations. It thus turns out that the robot succeeds in getting correctly re-localized after 10 time-steps, when the localization error returns to its initial value, typically equivalent to the robot's diameter (50 cm). The large augmentation of the error between the third and seventh time-steps is caused by perceptual aliasing that causes the environment near position B to look very similar to the environment near position C. Consequently, while the robot is effectively positioned near position B, the system wrongly estimates that there is a high probability of its being near position C. Such an incorrect inference gets corrected after 10 time-steps when the robot is far enough from position B for the environment to be sufficiently different from what it looks like near position C.

It is important to note, however, that, contrary to what was demonstrated in simulation in a previous paper (Filliat and Meyer, 2000), such a re-localization capacity may temporarily prove to be inefficient. The main reason is that the real vision system is much noisier than the simulated one, which enhances perceptual aliasing difficulties. As a consequence, information provided by sonar sensors and by idiothetic cues about the structure of the environment is assigned much greater importance in actual case than in simulation. This causes the re-localization procedure to become inefficient on the real robot when, for instance, a wrongly estimated position belongs to the same corridor as the real one. In this case, re-localization is not effective until the robot has entered an open area or a different corridor. Unfortu-

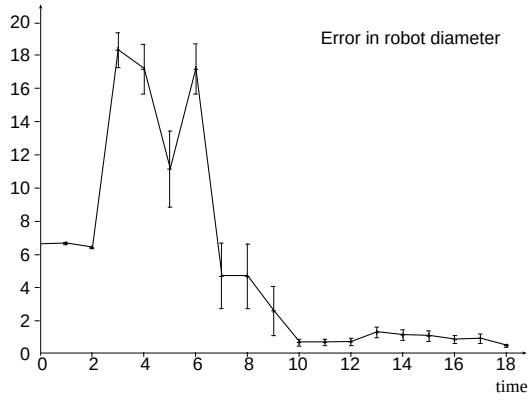

Figure 8: Evolution of the error in the estimation of the position after a passive displacement of the robot from point A to point B (Figure 5).

nately, the exploration strategies employed in the model emphasize strong local exploration in order to avoid localization errors. When such re-localization issues are encountered, local exploration prevents movements that would rapidly lead the robot out of a corridor and that would make prompt re-localization possible.

A solution to this problem would be to implement an active navigation strategy that would guide the robot toward areas where re-localization would be efficient. This suggestion is supported by the fact that, in the current system, manually assigning a goal to the robot when it is temporarily lost entails getting out of the corridor in question and permits a rapid re-localization.

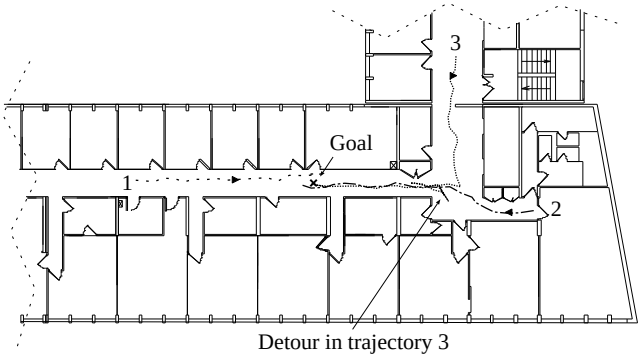

Figure 9: Three examples of goal-directed trajectories starting from three different positions. Trajectories 1 and 2 are direct, while trajectory 3 entails a re-planification leading to a slight detour.

Finally, the model makes it possible to efficiently reach any goal position in the environment. To demonstrate this, we performed ten trials to reach a fixed goal, starting from different positions. Among these trials, one failed due to the robot's getting trapped into a narrow dead-end. The nine other trials were successful, because the robot either directly reached the goal in five trials, or after the use of the above-mentioned re-planning procedure in the last four ones (Figure 9). The mean precision of the final positions in these nine trials was 50 cm, all trials ending less than 80 cm from the goal. These data are representative of the performance obtained with any other goal in the environment.

5 Discussion

The navigation system presented here therefore affords important autonomy capacities to mobile robots by combining global localization with map-learning. Its performances are achieved using relatively simple sensors and without resorting to metrical models for these sensors. The localization precision thus obtained (50 cm) is sufficient for most navigation tasks in common office environments[3]. In cases where it wasn't, the existing procedures could be supplied with additional short-range visual guidance algorithms, as demonstrated by current research efforts (Gourichon and Meyer, 2002).

The absence of any metrical model for the sensors is compensated in our model by the need for an exhaustive exploration of the environment. Indeed, the navigation system strongly relies on careful exploration to avoid localization instabilities during the map-learning process.

The capacities of the system have been demonstrated on a real robot in an environment mostly made up of hallways. Experiments in simulation indicate that navigating in open environments will be possible without any loss of precision using an omnidirectional camera. However, when a directional camera is used, as is the case in this paper, the system could present instabilities in the mapping process due to the higher rate of localization failures caused by the incompleteness of available data. In this case, the structure of the environment provided by the corridor is important, as shown by the mentioned limitations to the re-localization capacity. Further experiments in wider environments and using an omnidirectional camera will be conducted in the context of a new application within the AnimatLab, the Psikharpax project.

Complete autonomy of the system would be achieved if the robot were able to monitor its direction, along with its position. Indeed, the current method - which entails estimating the direction through odometry and periodically correcting the resulting error through an external reference - could be automated if the robot were able to learn how to associate the relative positions of some landmarks with its current orientation. Encouraging results have already been obtained with a preliminary implementation of such a capacity. This implementation entails first detecting colored landmarks from the initial position (using the method described in Gourichon and Meyer, 2002) and memorizing the directions

[3]For example, it allows a door to be reached correctly.

of these landmarks in the first map-node. The robot is then periodically guided by our navigation system toward this initial position where its direction estimate is reset using the perceived direction of these landmarks. Improvements to this scheme should entail memorizing such landmarks in several nodes of the map, so as to be able to reset the direction estimate in several positions and to avoid recurrent visits to the start node.

As mentioned in the previous section, the system could also be improved by the implementation of active navigation strategies to enhance the re-localization capacity. Such strategies could, for example, guide the robot, according to current position hypotheses, toward areas where the positions corresponding to the various hypotheses would be easy to differentiate.

With respect to other navigation models, this one shares several features with the ELAN model presented by Yamauchi and Beer (1996). However, the authors report that the latter model, which was functional in simulation, failed in real robot experiments. We believe that three main differences with respect to ELAN allow our model to work on a real robot and that they are therefore important for robustness :

- the regular correction of the direction by an external procedure that avoids large direction estimation errors and allows meaningful activity estimation,
- the use of vision instead of range sensors to reduce perceptual aliasing,
- the use of a dedicated heuristic to decide when to add a new node to the map.

This third point is particularly interesting, as the use of the heuristic mentioned by Yamauchi and Beer, i.e., a threshold on the most activated node, leads in our model to a severe loss of robustness. Comparisons of our model with other approaches can be found in Filliat and Meyer (2000).

Finally, this model is highly reminiscent of several biologically inspired navigation models described in the literature (Trullier et al., 1997). Indeed, nodes that have been used herein may be viewed as counterparts of *place-cells* found in the hippocampus of the rats. Our approach, however, relies on global localization, while most existing biologically-inspired models (e.g., Balakrishnan et al., 1999; Arleo and Gerstner, 2000) simply call upon single-hypothesis tracking and upon special procedures for the initial estimation of the position. Nevertheless, there are some indications that rats might in fact resort to global localization procedures also. For example, Zemel et al. (1997) describe a method to encode arbitrary probability distributions in the activities of a population of neurons. This technique potentially allows multiple-position hypotheses to be encoded in place-cell activities in a way very similar to what is done in our model. Another paper (Zhang et al., 1998), also demonstrates that deducing the position of a rat in a maze from place-cell recordings is much more precise when a probabilistic framework similar to that underlying this model is used, instead of resorting to a standard method like population vector coding.

In other words, such cues suggest that it might be useful to interpret the functioning of the hippocampus of rats during navigation within a probabilistic framework similar to the one used in this article.

6 Conclusion

The navigation system presented herein allows a high degree of autonomy by integrating global localization and map-learning processes with minimal human intervention. Moreover, this integration has been achieved using simple sensors, without resorting to any metrical sensor model, through the implementation of dedicated heuristics. Its capacities have been demonstrated on a real mobile robot operating in an unmodified office environment. Current research efforts to further enhance the autonomy of the system already provided encouraging results.

There is also good reason to think that the inner workings of the model could bear some resemblance to their biological counterparts found in the rat.

7 Acknowledgements

This work was supported by Robea, an interdisciplinary program of the French Centre National de la Recherche Scientifique.

References

Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation : A model of hippocampal place cell activity. *Biological Cybernetics, Special Issue on Navigation in Biological and Artificial Systems*, 83:287–299.

Balakrishnan, K., Bousquet, O., and Honavar, V. (1999). Spatial learning and localization in rodents : A computation model of the hippocampus and its implications for mobile robots. *Adaptive Behavior*, 7(2):173–216.

Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A. (1996). Acting under uncertainty : Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Dedeoglu, G., Mataric, M., and Sukhatme, G. S. (1999). Incremental, online topological map building with a mobile robot. In *Proceedings of Mobile Robots XIV - SPIE*.

Duckett, T., Marsland, S., and Shapiro, J. (2000). Learning globally consistent maps by relaxation. In *Proceedings of the International Conference on Robotics and Automation (ICRA'2000)*.

Filliat, D. (2001). *Cartographie et estimation globale de la position pour un robot mobile autonome (in french)*. PhD thesis, Université Pierre et Marie Curie.

Filliat, D. and Meyer, J. A. (2000). Active perception and map-learning for robot navigation. In *From Animals to Animats 6. Proceedings of the Sixth Conferences on Simulation of Adaptive Behavior*. The MIT Press.

Filliat, D. and Meyer, J.-A. (2002). Map-based navigation in mobile robots - I. A review of localization strategies. *Submitted for publication*.

Fox, D., Burgard, W., Thrun, S., and Cremers, A. B. (1998). Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National Conference on Articial Intelligence (AAAI-98)*.

Franz, M., Scholkopf, B., Georg, P., Mallot, H., and Bulthoff, H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125.

Gaussier, P., Joulain, C., Banquet, J., Lepretre, S., and Revel, A. (2000). The visual homing problem : An example of robotics/biology cross-fertilisation. *Robotics and autonomous systems*, 30(1-2):155–180.

Gourichon, S. and Meyer, J.-A. (2002). Using coloured snapshots for short-range guidance in mobile robots. *Special Issue on Biologically Inspired Robots - International Journal of Robotics and Automation*, Submitted for publication.

Meyer, J.-A. and Filliat, D. (2002). Map-based navigation in mobile robots - II. A review of map-learning and path-planing strategies. *Submitted for publication*.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angular sensors. In *Proceedings of the IEEE International Conference On Robotics and Automation (ICRA-85)*. IEEE Computer Society Press.

Piasecki, M. (1995). Global localization for mobile robots by multiple hypothesis tracking. *Robotics and Autonomous Systems*, 16:93–104.

Shatkay, H. and Kaelbling, L. P. (1997). Learning topological maps with weak local odometric information. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*.

Simmons, R. and Koenig, S. (1995). Probabilistic navigation in partially observable environments. In Mellish, S., (Ed.), *Proccedings of IJCAI'95*. Morgan Kaufman Publishing.

Smith, R., Self, M., and Cheeseman, P. (1988). Estimating uncertain spatial relationships in robotics. In Lemmer, J. F. and Kanal, L. N., (Eds.), *Uncertainty in Artificial Intelligence*, pages 435–461. Elsevier.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). MINERVA : A second generation mobile tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-1999)*.

Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2000)*.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55:189–208.

Trullier, O., Wiener, S., Berthoz, A., and Meyer, J. A. (1997). Biologically-based artificial navigation systems : Review and prospects. *Progress in Neurobiology*, 51:483–544.

Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2000)*.

Yamauchi, B. and Beer, R. (1996). Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part B,Special Issue on Learning Autonomous Robots*, 26(3):496–505.

Yamauchi, B., Schultz, A., and Adams, W. (1999). Integrating exploration and localization for mobile robots. *Adaptive Behavior*, 7(2):217–230.

Zemel, R. S., Dayan, P., and Pouget, A. (1997). Probabilistic interpretation of population codes. In Mozer, M. C., Jordan, M. I., and Petsche, T., (Eds.), *Advances in Neural Information Processing Systems*, volume 9, page 676. The MIT Press.

Zhang, K., Ginzburg, I., McNaughton, B., and Sejnowski, T. (1998). Interpreting neuronal population activity by reconstruction : A unified framework with application to hippocampal place cells. *Journal of Neurophysiology*, 79:1017–1044.

Interactive learning of visual topological navigation

David FILLIAT
ENSTA - UEI
32 boulevard Victor
75015 PARIS - France
david.filliat@ensta.fr

***Abstract*— We present a topological navigation system that is able to visually recognize the different rooms of an apartment and guide a robot between them. Specifically tailored for small entertainment robots, the system relies on vision only and learns its navigation capabilities incrementally by interacting with a user. This continuous learning strategy makes the system particularly adaptable to environmental lighting and structure modifications. From the computer vision point of view, the system uses a purely appearance-based image representation called bag of visual words, without any metric information. This representation was adapted to the incremental context of robotics and supplemented by active perception to enhance performances. Empirical validation on real robots and on the publicly available INDECS image database are presented.**

I. INTRODUCTION

Navigation is a fundamental capacity for mobile robots and numerous solutions have been proposed, adapted to different kind of robots. In this paper, we are specifically interested in small entertainment robots of humanoid or animal shape. Vision is the best suited sensor for these platforms due to its low cost, wide availability, low power consumption and highly informative output.

Vision-based navigation systems may use either topological or metrical maps [1]. In topological maps, only places such as rooms and their relations are learned and recognized [2], whereas in metrical maps, the precise positions of environment features and of the robot are estimated (e.g. [3]). In realistic scenarios for entertainment robotics, the robot is often moved directly by the user from one place to another, can fall or be blocked in places where sensors will have difficulty to find useful information (e.g. under tables, in corners...). In these situations, a metrical approach, that usually requires a continuous tracking of features, will probably fail, whereas a topological approach, able to recognize the rooms and guide the robot between them is more adapted. Moreover, topological approaches may be purely appearance based, thus avoiding the need for camera calibration.

In vision-based topological approaches, the use of a panoramic camera is common (e.g. [4], [5], [6]): this kind of sensor provides 360 information about the surroundings of the robot at one time, thereby making place recognition easier for example. In a humanoid or animal-like robot context, however, the use of a standard gaze-controlled camera is more natural, even if potentially more difficult to use. The introduction of active perception strategies [7], which are a key difference between computer vision and vision applied to robotics, is a natural way to compensate these difficulties.

The learning process should also be adapted to the context, where the user is usually eager to interact with his robot and is waiting for biologically plausible behaviours of the robot. It is therefore possible to take advantage of discontinuous user supervision to incrementally and progressively learn the navigation capabilities needed by the platform, instead of relying on a separate learning phase. As an interesting consequence, the space representation used by the robot will correspond to the concepts used by humans for navigation, thereby facilitating human-robot interactions. Such incremental training is also important to adapt the robot's spatial knowledge to the evolution of the environment such as varying lighting conditions and minor structure modifications (e.g. objects that are moved).

To provide a complete topological navigation system without using metric information, the system presented here integrates two components: a qualitative localization and mapping system and a visual homing method. The localization system (previously presented in [8]) is able to incrementally learn to recognize different rooms, while the visual homing method learns to guide the robot between rooms. Visual homing is a closed-loop strategy that iterates local goal direction prediction from an image and fixed length movement in the predicted direction. In this paper, we present a new evaluation of our qualitative localization method on the publicly available INDECS image database [9], and empirical evaluation of the visual homing method on real robots.

II. RELATED WORK

Using a standard camera, the authors of [9] perform qualitative localization by training a Support Vector Machine to predict the current room. Images are characterized by global histograms and the approach is shown to be robust over time to lighting and environment evolution. The approach has been adapted to incremental learning in [10]. The method proposed in [11] is based on scale-invariant visual keypoints to localize the robot through an image database representing the environment. Localization is performed by finding the image in the database that best match the current image. Robustness to lighting modifications is obtained by using temporal coherency of localizations. The system presented in [12] is using similar information in a two stage approach to enhance localization precision. These two systems rely on an a priori database describing the environment. However, the authors of [13] and [14] use similar approaches with online

acquisition of the image database, but localizing at the image level and not segmenting the environment at a higher level such as the rooms.

All these systems perform localization in a passive way, localizing the robot for each acquired image. However, in topological navigation, the current position is not modified by rotating the robot's camera. Conversely multiple images taken by moving the camera could be used for the estimation of the current position, as done in the work reported here. Active perception exploiting this property has been used with metric localization systems (e.g. [15]) but is not common in topological systems. To our knowledge, only [16] presents such an active localization scheme that searches for informative images to localize the robot, with a method similar to the one presented in this paper.

As in vision-based topological localization approaches, panoramic vision is often used to achieve visual homing ([17], [5]). However, in [18], while originally using panoramic vision, the authors report an adaptation with a standard camera without loss of performance, but requires an estimate of the robot's absolute direction by an external mean. Using a standard camera, most authors rely on metric information: for example, the system presented in [13] uses an estimation of geometric transformation between images to guide the robot. The research field of visual servoing also provides homing methods when applied to mobile robotics: using feature tracking and local 3D reconstruction between images, the authors of [19] control a robot to reproduce a path only specified by the image sequence acquired on this path. Few approaches however are not using metric information: [20] use a qualitative approach relying on feature tracking and qualitative control and [14] rely on image matching to choose the robot direction.

Finally, most of these methods either rely on supervised learning through an initial data acquisition phase ([9], [11], [12], [18], [16], [5]) or on autonomous segmentation of the environment ([14], [19], [13]). However, in a dynamic world, this initial or autonomous training cannot be guaranteed to provide robustness to environment and lighting modifications, and the ability to update the underlying model on-line, as proposed in this paper, is crucial. Some of these systems were therefore adapted to incremental learning with user supervision ([10]), by integrating small databases of new data. Our system integrates new user labelled data at a finer scale, requiring new data only when navigation is not possible. This is more similar to the concept of Human Augmented Mapping ([21], where only range sensing was used), where the robot incrementally discovers its environment guided by a human supervisor.

The main contribution of our work is therefore the integration of a purely appearance-based approach to qualitative localization at the level of rooms and visual homing to guide the robot between the recognized rooms. This is made possible using a standard perspective camera and without using any metric information by the coupling of active perception and incremental learning with user interaction.

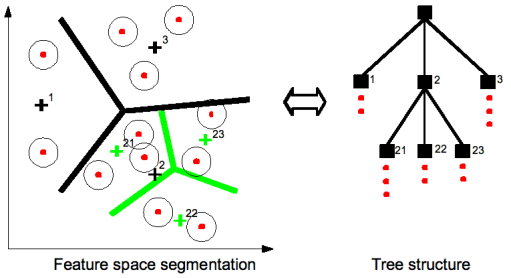

Fig. 1. Illustration of the tree structure with $k = 3$. Left: The crosses illustrate the node centres, the circles illustrate the words. Right: Only the word centres are stored in the leaves of the tree.

III. Incremental Bag of Words method for robotics

We adopted the popular "bag of visual words" approach to represent images in our system. Our contribution is an adaptation of this method in a purely incremental setup well suited for robotics, including the construction of a fast search structure for the visual words.

Bags of visual words is a popular method for image categorization [22] that relies on a representation of images as a set of unordered elementary visual features (the words) taken from a dictionary (or codebook). Using a given dictionary, a classifier is simply based on the frequencies of the words in an image, thus ignoring any global image structure. The term "bag of words" refers to document classification techniques that inspired these approaches where documents are considered as unordered sets of words. Several applications also exist for robotics (e.g. [12], [23]).

The words used in image processing are local image features such as SIFT keypoints (Scale Invariant Feature Transform) [24]. As these features are sensitive to noise and are represented in high dimension spaces, they are not directly used as words, but are categorized using vector quantization techniques such as k-means. The output of this discretization is the dictionary. Instead of building the dictionary off-line on an image database as is performed in most applications, we introduce an incremental dictionary construction ([8]) that makes it possible to start with an empty dictionary and build it as the robot discovers its surroundings. Our system therefore makes no a priori hypothesis on the type of environment it will face. The words in our system are balls of fixed radius in the feature space. Dictionary construction entails adding a new word centred on any feature that does not belong to an already existing word. The size of the balls is called the dictionary radius and influences the dictionary size, the algorithm performances and computation time (see [8]).

When using bag of words techniques with large vocabularies as is done in our system, searching for the word corresponding to a feature is a time consuming process. We therefore developed a tree dictionary structure to accelerate this operation. This structure is similar to that of [25], but built incrementally (figure 1). Each internal node of the tree has a set of k children, each defined by a centre in the feature

```
word_list = Search (node i, feature f)
  word_list = []
  if  (i is a leaf)
  | foreach word w in i
  | | if (dist(f,center(w)) < word_radius)
  | | | word_list = [word_list w]
  else
  | [s,d] = sort_children(f)
  | for j=1 to p
  | | if (d(j) < word_radius)
  | | | word_list = [word_list Search(s(j), f)]
  return word_list
```

Fig. 2. Search algorithm pseudo code. $dist(f, g)$ computes the distance between two features f and g. $word\_radius$ is the size of the words in the dictionary. $s, d = sort\_children(f)$ returns the list s of children sorted according to the distance of their frontier to feature f and the corresponding list of distances d. p is the maximum number of children to consider. See text for details.

space. Each child stores the word centres that are the closest to its centre, thus partitioning the feature space of the parent node by the Voronoi diagram of the k children centres.

The building process is fully incremental and simply begins with an empty root node. Any new word that should be added to the dictionary is directly added to the leaf node to which its centre belongs. If the number of words stored in this leaf is above a threshold n_w, the leaf is split in k children. The centres of the children leaves are defined by applying k-means to the n_w words centres. We applied this procedure in the experiments of this paper with $n_w = 500$ and $k = 10$. Although this procedure does not enforce a balanced structure to the tree[1], therefore potentially penalizing the search efficiency, experimental results show that the trees are always nearly balanced with a depth variation among branches of less than 2 and a limited impact on search speed.

As shown by Beis and Lowe [26], searching for words in these structures in high dimension (e.g. 128 for SIFT descriptors) leads to a complexity similar or even worse than that of naive linear search because a large number of nodes is examined, thus compromising any interest in the use of a tree structure. This scaling problem was solved in [26] in the case of kd-trees by the design of a fast approximate search procedure. We use a similar method, by limiting the number of children to be explored in each node to $p < k$ and by searching first in the children whose frontiers are the closest to the searched feature (Figure 2). This procedure affords a very fast search – at the cost of a low percentage of errors. For example, in the experiments reported in this paper, the search for the words corresponding to a SIFT feature in a dictionary of 15000 words with $p = 3$ took in average 1.4 ms with an error rate of around 0.6%.

This search procedure rely on the use of L2 distance for the calculation of the distance to the node frontier. However, in some cases, the use of another distance is preferred. For example, color histograms are better compared using the diffusion distance [27] we use in this paper (see below). As this distance does not stem from a dot product, rapidly calculating the distance between a feature and a node frontier is not possible. It is therefore not possible to estimate if a neighboring node has to be searched or not. For these cases, we devised another approximate search strategy that *exhaustively* explores a given number $q < k$ of children for each node, starting with the children whose *centres* are the closest to the feature. Experiments with diffusion distance and color histograms show that this procedure leads to a small search time with very few errors. In the experiments reported in this paper, the search for the words corresponding to a H histogram feature, in a 15000 words dictionary with $q = 3$ took around 2.0 ms with an error rate of 0.6%.

As shown in [8], performances can be improved by integrating several feature spaces. To this end, a dictionary is built for each feature space, and the classifiers integrate the words taken from all the dictionaries (see next section). In this paper, two feature spaces using complementary image characteristics were used:

- SIFT keypoints [24]: interest points are detected as the maximum over scale and space of the convolution by differences of Gaussian. Keypoints are described by histograms of gradient orientations around the detected point and are invariant in scale and rotation. The descriptor used are of dimension 128 and are compared using L2 distance.
- Local color histograms: The image is decomposed in a set of overlapping windows of several sizes in order to provide some scale invariance. The histograms of the H value in the HSV color space for each window are used as features. The windows used are of size 40x40 pixels taken each 20 pixels and 20x20 pixels taken each 10 pixels. The descriptors are of dimension 16 and are compared using diffusion distance [27].

IV. System overview

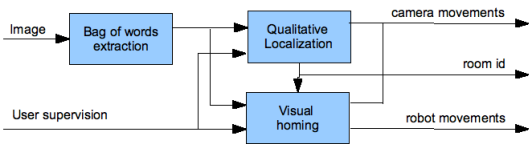

Fig. 3. Functional overview of the system.

Our navigation method uses the same bag of words image representation for qualitative localization and visual homing (figure 3). A module therefore transforms images coming from the robot's camera into their bag of words representations, incrementally building the corresponding dictionaries. These representations are used by the localization module to predict the room identity and by the visual homing module to estimate the direction from the current room (estimated by the localization module) to the room requested by the user. In these two modules, the classifiers should be trained incrementally, i.e. they should be able to process new examples and add new categories without the need to reprocess all the previous data. To achieve that, we used voting methods in which training simply entails updating word statistics, and classifying simply entails reading these statistics.

[1] as is usually required in kd-trees for example

As will be detailled in the next subsections, the localization and visual homing modules both use active perception strategies, potentially requesting new images with a different camera orientation to perform their task. These two modules also perform learning incrementally using discontinuous supervision from the user. The user can provide the room identity to the localization module at any time, while the correct goal direction is requested by the visual homing module.

A. Localization and mapping module

The map in our approach is composed of statistics associated to the visual words, i.e. the rooms in which each word has already been seen in the examples used for training.

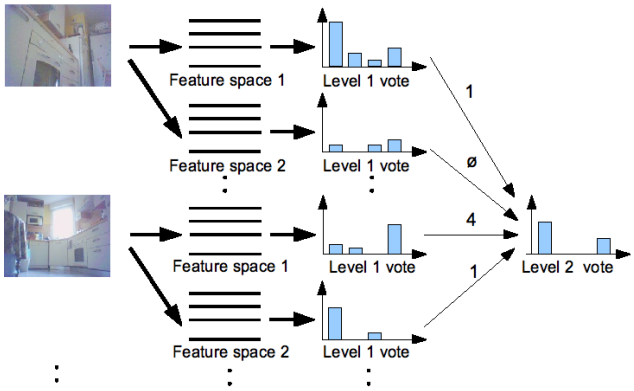

Fig. 4. Illustration of the two stage voting method used for qualitative localization.

A two stage voting method implementing the active perception procedure is used to estimate the robot position (figure 4). In a given position, a first picture is taken from the current head direction. The words found in the image vote at a first level for the rooms in which they have already been seen. Each word votes using its normalized *inverse document frequency*, giving more weight to the words that correspond to fewer locations:

$$idf = log(N/n_i)/log(N)$$

where N is the total number of rooms and n_i the number of rooms in which the word i has been seen.

A quality of the vote result is calculated as the relative difference between the maximum and the second maximum:

$$quality = \frac{v_{Winner} - v_{Second}}{\sum_j v_j}$$

where v_j is the number of votes for room j.

In order to filter out non-informative images that bring noise in the estimation, the winning room votes at the second level (with its quality) only if the quality is above a threshold, 0.1 in this paper (see [8] for an evaluation of the threshold influence).

This process is repeated with the other feature spaces and with new images until the quality of the second level vote (estimated with the same method) reaches a given threshold (0.5 in all experiments) or a given number of images is reached (5 in all experiments). The recognized room is then the room with the highest score. The new images taken for localization are taken with a new random head direction without moving the robot's body.

The associated mapping procedure is interactive and processes images upon user feedback after the localization procedure is performed. If the user declares the localization incorrect, learning is performed using the room label given by the user. Images that have been used for localization and new images taken from random head directions are used for learning (for a total of 10 images in the experiments reported). Learning these images entails simply memorizing that the corresponding words have been seen in the current room. The succession of localization events in different rooms, at different positions and under different lighting condition, learning when errors are committed, eventually converges to a correct representation of rooms and to stabilization of the recognition performances (see Results section).

B. Visual homing module

This module learns, for each room, several visual homing strategies that can guide the robot to the different neighboring rooms. A homing strategy makes it possible to infer the local direction to take to reach the goal from any position in the room. Goal reaching is performed by iterating predictions of the goal direction from the current camera image and movement of the robot in this direction for a fixed distance.

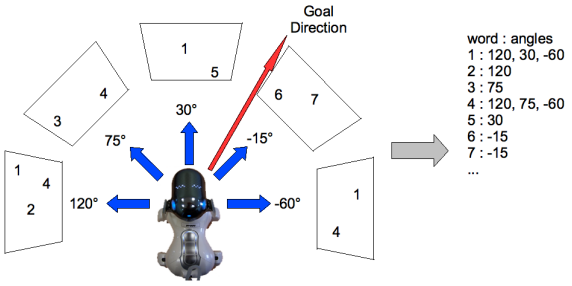

Fig. 5. Illustration of the visual homing learning procedure.

The learning procedure is triggered each time it is not possible to predict the goal direction. The procedure first asks the user for the local goal direction. Five images are then captured by moving the robot head from one side to the other. The visual words from each image are associated with the relative direction between the robot head and the goal (figure 5). A homing strategy is therefore memorized as a list of angles for each word in the dictionaries. For each word, the mean and standard deviation of the associated angles are estimated.

Predicting the goal direction from an image is performed using a voting method. The directions around the robot are discretized with a step of 20 degrees. Each word found in the image vote for the bin corresponding to its mean associated direction. Words which are found in different parts of the

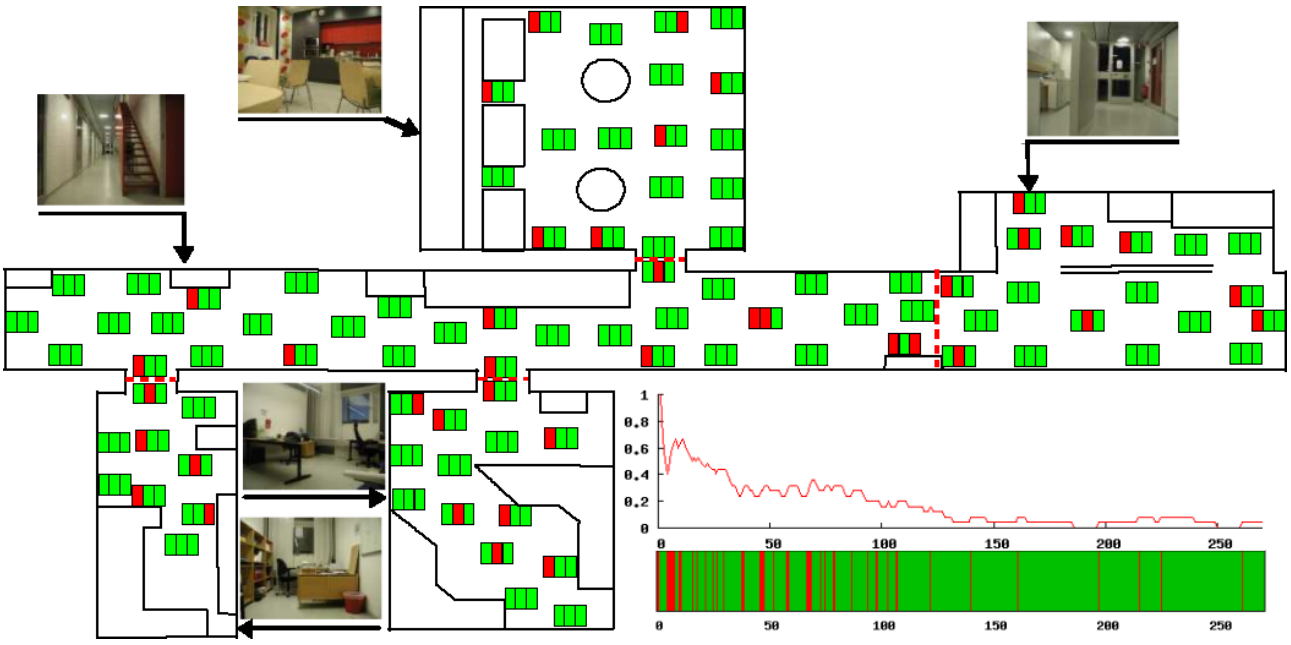

Fig. 6. Example of localization results on INDECS database. For each position, 3 rectangles show the 3 successive localization experiments performed at this point with different lighting conditions. A green rectangle corresponds to a successful localization, a red one corresponds to an error and the learning of the position for this lighting condition. The diagrams at the bottom right show the temporal succession of correct and incorrect localization results (bottom) and the evolution of the error rate on the last 25 localizations (top).

environment (e.g. words 1 and 4 in figure 5) are excluded from the vote through a threshold on the standard deviation of their associated directions. In our experiments, words with a standard deviation of more than 20 degrees were excluded. If the quality of the vote result (estimated as in the localization module) is below a threshold (0.1 in the experiments), an active perception procedure requests new images by turning the robot head 45 degrees to the left and to the right. If none of these images produce a vote with a sufficient quality, the learning procedure is performed. Otherwise, the robot is turned in the predicted goal direction, and is moved forward by a fixed distance (50cm in our experiments) before performing the procedure again.

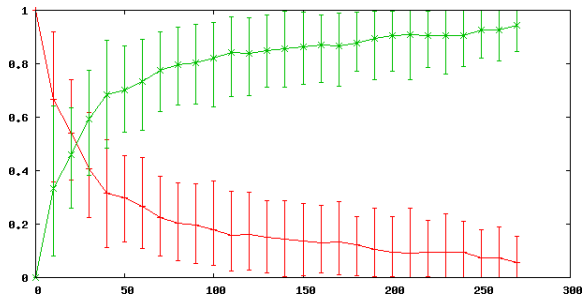

Fig. 7. Evolution of the error rate (red) and correct localization rate (green) during the last 25 localizations. The graph shows the mean values on 100 random experiments similar to the one presented in figure 6, with minimum and maximum values plotted as error bars.

V. Experimental results

A. Localization

In a previous article [8], we evaluated our localization method on a Sony Aibo robot. We present here new validation results on the publicly available INDECS database [9]. This database contains images taken at 91 different points and under three different lighting conditions (sunny, cloudy, night) in an environment made of five different rooms (figure 6). For each of the 271 positions[2] present in the database, 12 images were taken by rotating horizontally the camera of 30 degrees between images.

Evaluations were conducted by taking the 271 positions in random order, thus mimicking the incremental discovery of the environment by the robot at different positions and different time. The localization algorithm was applied for each position and learning was performed when an error was made. Figure 6 gives an example of such an evaluation sequence. In this example, 42 positions out of the 271 needed to be learned. We can see that the positions where learning was required are scattered across the whole environment, thereby naturally covering the different viewpoints in the environment. The fact that learning is most of the time performed less than one time for each point in the environment also demonstrates the robustness of our approach to lighting conditions and to minor modifications such as the presence or absence of people. The frequency of localization errors also decreases to around 4% after the first 100 localizations.

Performing these evaluations 100 times with different random position order shows that the mean number of learning

[2]We will call position a point with an associated lighting condition, thereby considering a point with different lighting as different positions where our system can try to recognize the room

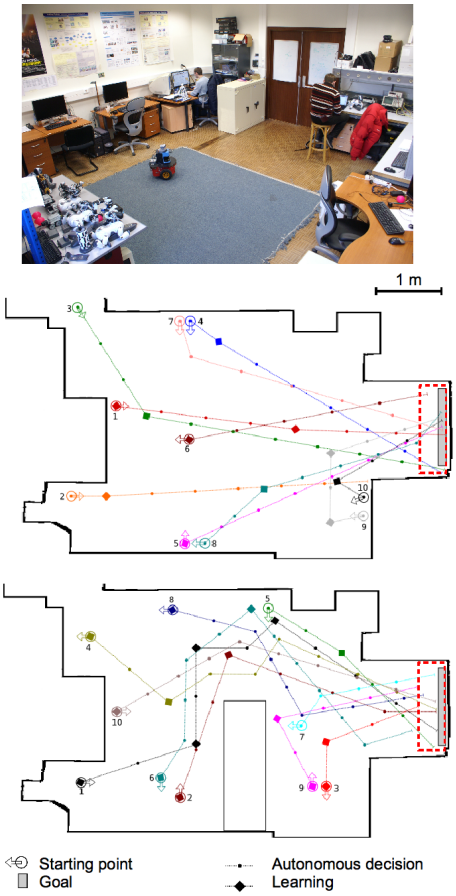

Fig. 8. Visual homing results in the main room of our laboratory, with and without a central obstacle. The goal is the dashed rectangle on the right.

event is 52.3, meaning that globally, 80% of the positions are correctly recognized. Figure 7 shows the evolution of the local rate of correct localization, showing that the localization accuracy continuously increases, reaching a level of 90% after 180 localization event. The level of 80% of correct recognition is reached after 100 localizations, with a mean of 30 trainings, corresponding to 6 positions learned in each room. Comparing these results with the ones presented in [28] (where the best classification rate was 81% on the whole dataset images with support vector machines) shows the advantage of using an active perception strategy. By automatically discarding uninformative images, and by recognizing the positions instead of all the images, our system is able to more efficiently recognize the different robot positions with a simpler machine learning algorithm. Moreover, our system only uses 20% of the images for learning instead of 33% used in [28].

B. Visual homing

Visual homing was validated in the same environment on a Sony Aibo[3] and on a MobileRobots Pioneer 3 dx. Performances were similar on the two robots. Figure 8 shows two examples of training sessions in our lab with the Pioneer 3 dx robot. The first trial was made in an open environment, with people working at their desks on the periphery. In the second, we added a large obstacle in the centre of the room to validate more complex homing strategies. In both experiments, the goal was the exit door of the room. The homing strategy was considered successful when the robot reach a 40cmx1m rectangle in front of the door. The robot was asked to reach this goal from 10 different starting positions with different orientations. In the first setup, 11 user supervisions were necessary for the robot to learn to reach the goal from all positions. In the second, more complex, setup, 15 were necessary.

In both experiments, the final homing strategy is able to guide correctly the robot to the exit door of the room. During learning, the precision obtained for the final point is low when the starting points are varied, with an error between the end-point and the rectangle centre reaching 50 cm in some experiments, but is sufficient for the robot to exit the room. The learning points are scattered in the environment, showing that the homing strategies are correctly learned as the robot is able to predict the goal direction for positions close to previous learning points. After learning, when repeating a trajectory from a given starting point, the error is smaller: 10 trials, starting from position 10 in the second environment of figure 8 leads to a mean error of 15 cm, with a maximum error of 30 cm.

VI. Discussion

Thanks to the active perception strategy, the overall performances obtained for localization are correct, using a simple appearance-based model with a perspective camera and simple learning algorithms. A limitation is the variance of the obtained results (figure 7): results can be very good (i.e. 95% of correct recognition after 100 localizations) when the user chooses correctly the localization positions, i.e. positions in open areas that rapidly covers the whole environment. But results can be bad when positions are not well chosen (i.e. less than 80% of correct recognition at the end of the experiment). However, in a realistic scenario, users have a natural tendency to guide the robot in central and open areas of the rooms, where the performances of our method are the best.

Compared to autonomous topological navigation in a similar setup ([14]), labelling places and learning homing behavior by interaction with the user has the advantage of adapting the space segmentation on-line by asking supervision to the user when the robot encounters an ambiguous viewpoint. This can be viewed as an active learning strategy, where only relevant examples are used for learning. The consequences are that less examples are required than in supervised settings [29] and that the method is stable in the long term as learning is not performed once performances are correct. A potential problem arises when a user makes errors in supervision, or tries to make the robot differentiate very similar rooms or parts of a room. In our system, this will lead to ever more requests for learning and an eventual

[3]Video available at http://cogrob.ensta.fr/indoornavigation.html

permanent confusion of the rooms. Statistics on the visual words could be analysed to warn the user in such cases.

The end point precision of the visual homing strategy is low, but is sufficient to reach a door in order to exit a room. The poor precision is linked to the fact that, contrary to more precise approaches ([13], [18], [19], [20]), only the appearance of images are used, without any metric information extraction. Our strategy also does not depend on the robot's odometry and does not require an external estimate of the robot orientation. A positive consequence is that this strategy can be used on simple platforms with low quality camera such as the Aibo robot, even with a very weak precision of movement execution. The question of when to stop a homing strategy is also important. In the experiments reported, the user stops the homing behavior when the robot reaches the door. In a more autonomous setup, localization should be attempted when homing is not possible so as to stop homing behavior if the target room is reached, or ask for user supervision otherwise.

From an implementation perspective, our system does not currently integrate planning capabilities, meaning that it can only guide the robot from one room to the neighboring ones. Integration of a complete topological map and chaining of homing strategies to go from one room to the other through a third one is the subject of future work.

VII. Conclusion

We have presented a visual topological navigation system adapted to small robots. The two modules designed to recognize rooms and guide the robot between rooms rely only on the appearance of images, without using any metric information. This simple representation is built in a fully incremental process, complemented by active perception strategies and user supervision for the learning of the navigation capabilities, making it possible to achieve efficient topological navigation on simple robots with standard perspective cameras.

Acknowledgment

The author would like to thank Jose-Luis Susa and Florian Vichot for their contribution to the implementation of the system presented here.

References

[1] D. Filliat and J. A. Meyer, "Map-based navigation in mobile robots - I. a review of localisation strategies," *Journal of Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.

[2] B. J. Kuipers and Y. T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.

[3] A. J. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007.

[4] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE Press, 2000, pp. 1023–1029.

[5] T. Goedeme, T. Tuytelaars, G. Vanacker, M. Nuttin, and L. V. Gool, "Omnidirectional sparse visual path following with occlusion-robust feature tracking," in *Proceedings of The sixth workshop on omnidirectional vision camera networks and non-classical cameras (OMNIVIS)*, 2005.

[6] O. Booij, Z. Zivkovic, and B. Kröse, "Sparse appearance based modeling for robot localization," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.

[7] Y. Aloimonos, *Active perception*, Y. Aloimonos, Ed. LEA, 1993.

[8] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2007.

[9] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06)*, Beijing, China, October 2006.

[10] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, "Incremental learning for place recognition in dynamic environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS07)*, 2007.

[11] J. Kosecka and X. Yang, "Location recognition and global localization based on scale invariant features," in *Wokshop on Statistical Learning in Computer Vision, European Conference on Computer Vision*, 2004.

[12] J. Wang, R. Cipolla, and H. Zha, "Vision-based global localization using a visual vocabulary," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[13] F. Fraundorfer, C. Engels, and D. Nistr, "Topological mapping, localization and navigation using image collections," in *In Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2007.

[14] E. Motard, B. Raducanu, V. Cadenat, and J. Vitri, "Incremental online topological map learning for a visual homing application," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[15] J. M. Porta, B. Terwijn, and B. Krse, "Efficient entropy-based action selection for appearance-based robot localization," in *Proceedings of the International Conference on Robotics and Automation ICRA'03*, 2003, pp. 2842–2847.

[16] F. Schubert, T. Spexard, M. Hanheide, and S. Wachsmuth, "Active vision-based localization for robots in a home-tour scenario," in *Proceedings of International Conference on Machine Vision Applications*, 2007.

[17] A. Vardy and R. Moller, "Biologically plausible visual homing methods based on optical flow techniques," *Connection Science*, vol. 17, pp. 47–89, 2005.

[18] C. Giovannangeli, P. Gaussier, and G. Dsilles, "Robust mapless outdoor vision-based navigation," in *IEEE/RSJ International Conference on Intelligent Robots and systems*, 2006.

[19] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'07*, 2007.

[20] Z. Chen and S. T. Birchfield, "Qualitative vision-based mobile robot navigation," in *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[21] E.A.Topp and H.I.Christensen, "Topological modelling for human augmented mapping," in *In Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2006.

[22] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV04 workshop on Statistical Learning in Computer Vision*, 2004, pp. 59–74.

[23] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Real-time visual loop-closure detection," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2008.

[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[25] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR06*, 2006.

[26] J. S. Beis and D. G. Lowe, "Indexing without invariants in 3d object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 1000–1005, 1999.

[27] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[28] A. Pronobis, "Indoor place recognition using support vector machines," Master's thesis, 2005.

[29] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, November 2001.

IEEE TRANSACTIONS ON ROBOTICS, SPECIAL ISSUE ON VISUAL SLAM, OCTOBER 2008 1

A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words

Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer

***Abstract*—In robotic applications of visual simultaneous localization and mapping techniques, loop-closure detection and global localization are two issues that require the capacity to recognize a previously visited place from current camera measurements. We present an online method that makes it possible to detect when an image comes from an already perceived scene using local shape and color information. Our approach extends the bag-of-words method used in image classification to incremental conditions and relies on Bayesian filtering to estimate loop-closure probability. We demonstrate the efficiency of our solution by real-time loop-closure detection under strong perceptual aliasing conditions in both indoor and outdoor image sequences taken with a handheld camera.**

***Index Terms*—Loop-closure detection, localization, SLAM.**

I. Introduction

Over the last decade, the increase in computing power has helped to supplement traditional approaches to simultaneous localization and mapping (SLAM [1], [2], [3], [4]) with the qualitative information provided by vision. As a consequence, in robotics research, commonly used range and bearing sensors such as laser scanners, radars and sonars tend to be associated with, or replaced by, single cameras or stereo-camera rigs. For example, in previous work [5], we performed vision-based 2D SLAM for Unmanned Aerial Vehicles (UAV). Likewise, in [6], the authors performed 3D SLAM in real-time at 30Hz using a monocular handheld camera, while the authors of [7] present visual SLAM solutions based on both monocular and stereo vision.

However, there are still difficulties to overcome in robotic vision in general, and in SLAM applications in particular. Among them, the loop-closure detection issue concerns the difficulty of recognizing already mapped areas, while the global localization issue concerns the difficulty of retrieving the robot's location in an existing map. Those problems can be addressed by detecting when the robot is navigating through a previously visited place from local measurements. The overall goal of the research effort reported in this article is thus to design a vision-based framework tackling these issues so as to make it possible for a robot to reinitialize a visual 3D-SLAM algorithm like one of those presented in [6] or [7] in such situations. This comes down to an online image retrieval task that consists in determining if current image has been taken from a known location. Such task bears strong similarities with image classification methods like those described in [8] and [9], but an important difference is our commitment to online processing.

In this paper, we present a real-time vision-based method to detect loop-closures in a Bayesian filtering scheme: at each new image acquisition, we compute the probability that the current image comes from an already perceived scene. To this end, we designed a scene recognition framework that relies on an incremental version [10] of the bag-of-words method [9]. Loop-closure hypotheses whose probability is above some threshold are confirmed when a coherent structure between the corresponding images is found - i.e. when the epipolar geometry constraint is satisfied. This ultimate validation step is accomplished using a multiple-view geometry algorithm similar to the one proposed in [11]. We provide experimental results demonstrating the quality of our approach by performing loop-closure detection in incremental and real-time conditions in both indoor and outdoor image sequences using a single monocular camera.

In section 2, we present a review of related work on visual loop-closure and global localization. Section 3 briefly introduces our implementation of the bag-of-words paradigm. The filtering scheme is detailed in section 4 and experimental results are given in section 5. The last two sections are devoted to discussion and conclusion.

II. Related Work

The Monte Carlo Localization (MCL) method was originally designed [12] to make global localization capitalizing on range and bearing sensors possible. Although successfully adapted to vision [13], this method does not match our requirements since it relies on the existence of a map obtained beforehand. From the same principle, the Rao-Blackwellised particle filter (RBpf) enables loop-closure capabilities in SLAM algorithms (e.g the FastSLAM [14] framework). It has also been adapted to vision [15], but it suffers degeneration when closing a loop due to inaccurate resampling policies [3]. In addition, RBpf are not loop-closure detection methods per se, but rather SLAM methods robust to loop-closure events.

Loop-closure detection has also been performed using an Extended Kalman Filter (EKF) application to visual SLAM ([16], [17]). The overall idea is to detect loop-closures from advanced data association techniques that try to match visual features found in current images with those stored in the map. This approach limits the information used to detect loop-closure to the information used for mapping (which is designed

Adrien Angeli (corresponding author), Stéphane Doncieux and Jean-Arcady Meyer are with the Université Pierre et Marie Curie - Paris 6, FRE 2507, ISIR, 4 place Jussieu, F-75005 Paris, France. Email: firstname.lastname@isir.fr

David Filliat is with Ecole Nationale Supérieure des Techniques Avancées, 32, bvd Victor, F-75015 Paris, France. Email: david.filliat@ensta.fr

Regular Paper

for SLAM, and not optimized for loop-closure detection). It is also linked to a particular SLAM algorithm, whereas our approach may be adapted to any SLAM method (even not vision-based).

In this work, we wish to design a simple visual system able to perform loop-closure detection and global localization, within the framework of an online image retrieval task. Following a similar approach, but in a non-incremental perspective, voting methods presented in [18] and [19] call upon maximum likelihood estimation to match the current image with a database of images acquired beforehand. The likelihood depends upon the number of feature correspondences between the images, and leads to a vote assessing the amount of similarity. In [18], the authors also use multiple-view geometry to validate each matching hypothesis, while in [19] the accuracy of the likelihood is qualitatively evaluated in order to reject outliers. Even though they are easy to implement, the aforementioned voting methods rely on an offline construction of the image database and need expensive one-to-one image comparisons when searching for the most likely hypotheses. Moreover, the maximum likelihood framework is not suitable for managing multiple hypotheses over time, as it does not ensure the time coherency of the estimation (i.e. information from past estimates is not integrated over time so as to be fused with actual ones). As a consequence, this framework is prone to transient detection errors, especially under strong perceptual aliasing conditions.

In [20] and [21], bag-of-words methods are used to perform global localization and loop-closure detection in an image classification scheme (see also [22] for an extended version of [21], with multi-robot map-joining addressed as a loop-closure problem). Bag-of-words methods ([8], [9]) rely on a representation of images as a set of unordered elementary features (the visual words) taken from a dictionary. The dictionary is built by clustering similar visual descriptors extracted from the images into visual words. Using a given dictionary, image classification is based on the occurrence of the words in an image to infer its class. In [20] and [21], images are represented as vectors of visual words' statistics with size equal to the number of words in the dictionary. The dictionary is built beforehand in an offline process, clustering the visual features extracted from a training database of images into representative words of the environment. Matching between current and past images is defined as a Nearest Neighbor (NN) search among the cosine distances separating the corresponding vectors. In [20], a simple voting scheme selects the n best candidates from the NN search and multiple-view geometry is used to discard outliers. In [21], the NN search results are used to fill a *similarity matrix* whose off-diagonal elements represent loop-closure events, thus providing a powerful way to manage multiple hypotheses. In both approaches, the use of a dictionary enhances the robustness of matches, enabling a good tolerance to image noise, but the NN search involved, relying on exhaustive one-to-one vector comparisons, is very expensive.

More recently, the authors of [23] have proposed a vision-based probabilistic framework that makes it possible to estimate the probability that two observations originate from the same location. This approach, based on the bag-of-words scheme, is very robust to perceptual aliasing: a generative model of appearance is learned in an offline process, approximating the probabilities of co-occurrences of the words contained in the offline-built dictionary. Using this model, loop-closure detection can be performed with a complexity linear in the number of locations. The main asset of this model is its ability to evaluate the distinctiveness of each word, thus accounting for perceptual aliasing at the word level, while its principal drawback lies in the offline process needed for model learning and dictionary computation.

In the majority of the methods presented above, SIFT (Scale Invariant Feature Transform [24]) features are the preferred input information because of their robustness to reasonable 2D affine transformations, scale and viewpoint changes. However, other visual features could be used for loop-closure detection and global localization (see [25] for a comparison of visual local descriptors). For example, as stated in [19], color histograms are powerful features providing a compact geometry-less image representation that exhibits some attractive invariance properties to viewpoint changes. Hence, it may be suitable to merge several complementary visual information, like shape and color for example, in order to obtain a reliable solution in different contexts.

III. Visual Dictionary

The implementation of the bag-of-words method used here is detailed in [10]: the dictionary construction is performed online along with the image acquisition, in an incremental fashion. The words are stored using a tree structure (see [26] for more details), enabling logarithmic-time complexity when searching for a word and thereby entailing real-time processing. In the work reported here, we used two different feature spaces to describe the images:

- SIFT features [24]: interest points are detected as maxima over scale and space in differences of Gaussians convolutions. The features are memorized as histograms of gradient orientations around the detected point at the detected scale. The corresponding descriptors are of dimension 128 and are compared using L2 distance.
- Local color histograms: the image is decomposed in a set of regularly spaced windows of several sizes to improve scale invariance. The normalized H histograms in the HSV color space for each window are used as features. The windows used here are of size 20x20 (respectively 40x40) taken every 10 (respectively 20) pixels. The descriptors are of dimension 16 and are compared using diffusion distance [27].

A dictionary is built for each feature space.

IV. Bayesian Loop-Closure Detection

In this paper, we address the problem of loop-closure detection as an image retrieval task: we are seeking for the past image, if it exists, that looks similar enough to the current one to consider that they come from close viewpoints. The overall processing, illustrated in the diagram of figure 1, is achieved in a Bayesian filtering framework estimating the probability

that current and past images pertain to the same scene: we thus look for the past image that maximizes the probability of loop-closure with the current image. When such an image is found (i.e. when probability is high for a particular loop-closure hypothesis), the consistency of the structure underlying those two images is checked by a multiple-view geometry algorithm [11]. When perceptual aliasing is present in the environment (i.e. when different places look similar), epipolar geometry provides a powerful way to reject outliers (i.e. past images that look like the current image but do not come from the same scene). In order to take advantage of different types of information, several feature spaces (i.e. SIFT features and H histograms) are used here for representing the images. Compared to maximum likelihood methods, the Bayesian filtering scheme proposed here takes temporal coherency of image acquisition into account in order to bring robustness to transient detection errors.

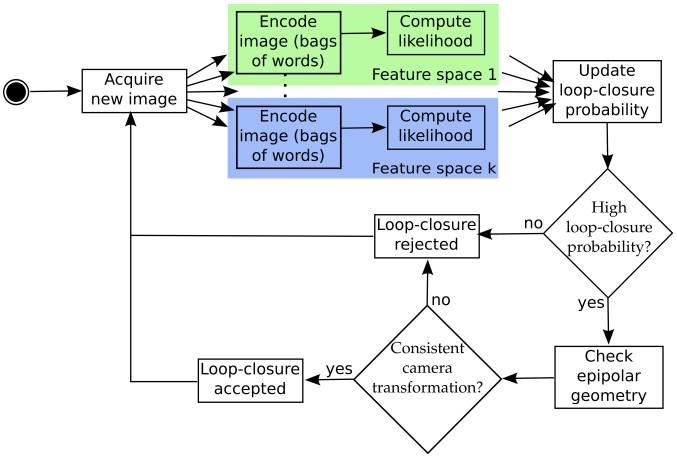

Fig. 1. Overall processing diagram (see text for details).

In this section, we first give the mathematical derivation of the filtering scheme used for the estimation of loop-closure probability. Then, we focus on issues regarding temporal coherency, likelihood computation and hypotheses management.

A. Discrete Bayes Filter

Let S_t be the random variable representing loop-closure hypotheses at time t. The event $S_t = i$ is the event that current image I_t "closes the loop" with past image I_i. This implies that the corresponding viewpoints x_t and x_i are close, and that I_t and I_i are similar. The event $S_t = -1$ is the event that no loop-closure occurred at time t. In a probabilistic Bayesian framework, the loop-closure detection problem can hence be formulated as searching for the past image I_j whose index satisfies:

$$j = \text{argmax}_{i=-1,\ldots,t-p} p(S_t = i|I^t) \tag{1}$$

where $I^t = I_0, \ldots, I_t$, with $j = -1$ if no loop-closure has been detected. This search is not performed over the last p images because I_t always looks similar to its neighbors in time (since they come from close locations), and doing so would result in loop-closure detections between I_t and recently seen images (i.e. $I_{t-1}, I_{t-2}, \ldots, I_{t-(p+1)}$). This parameter, set to 10 in our experiments, is adjusted depending on the frame rate and on the velocity of camera motion.

We therefore need to estimate the *full posterior*, $p(S_t|I^t)$ for all $i = -1, \ldots, t-p$, in order to find, if a loop-closure occurred, the corresponding past image.

Following Bayes' rule and under the Markov assumption the posterior can be decomposed into:

$$p\big(S_t|I^t\big) = \eta p\big(I_t|S_t\big) p\big(S_t|I^{t-1}\big) \tag{2}$$

where η is the normalization term. Let $(Z_k)_i$ be the state of the dictionary associated with the feature space k (SIFT features or H histograms in this paper) at time index i. The time subscript i is inherent to the incremental aspect of the dictionary construction: $(Z_k)_0 \subseteq (Z_k)_1 \subseteq \ldots \subseteq (Z_k)_{i-1} \subseteq (Z_k)_i$, with $(Z_k)_0 = \emptyset$ (features from the feature space k extracted in I_i are used to build $(Z_k)_{i+1}$). Also, let the subset $(z_k)_i$ of words taken from $(Z_k)_i$ and found in image I_i denote one representation of this image: $I_i \Leftrightarrow (z_k)_i$, with $(z_k)_i \subseteq (Z_k)_i$. Since several feature spaces are involved here, several image representations exist (one per feature space). Thus, let $(z^n)_i$ be the overall representation of image I_i, all feature spaces $k = 0, \ldots, n$ combined. The sequence of images I^t acquired up to time t can therefore be represented by the sequence $(z^n)^t = (z^n)_0, \ldots, (z^n)_t$.

So, the full posterior, now rewritten $p\big(S_t|(z^n)^t\big)$, can be expressed as follows:

$$p\big(S_t|(z^n)^t\big) = \eta p\big((z^n)_t|S_t\big) p\big(S_t|(z^n)^{t-1}\big) \tag{3}$$

Assuming independence between the feature spaces, we can derive a more tractable mathematical formulation for equation 3 so as to make computation of the full posterior easier. However, capturing the correlations existing between the different dictionaries could provide additional information about the occurrence of the words. Under the independence assumption, the full posterior's expression can be written:

$$p\big(S_t|(z^n)^t\big) = \eta \left[\prod_{k=0}^{n} p\big((z_k)_t|S_t\big)\right] p\big(S_t|(z^n)^{t-1}\big) \tag{4}$$

where the conditional probability $p\big((z_k)_t|S_t\big)$ is considered as a likelihood function $\mathcal{L}\,(S_t|(z_k)_t)$ of its second argument (i.e. S_t), with its first argument (i.e. $(z_k)_t$) held fixed: we evaluate, for each entry $S_t = i$ of the model, the likelihood of the currently observed words $(z_k)_t$ (see section IV-C).

Recursive estimation of the full posterior is made possible by decomposing the right hand side of equation 4 as follows:

$$p\big(S_t|(z^n)^t\big) = \eta \left[\prod_{k=0}^{n} p\big((z_k)_t|S_t\big)\right] \underbrace{\sum_{j=-1}^{t-p} p\big(S_t|S_{t-1}=j\big) p\big(S_{t-1}=j|(z^n)^{t-1}\big)}_{\textit{belief}} \tag{5}$$

where $p\big(S_t|S_{t-1}\big)$ is the time evolution model (see section IV-B) of the probability density function (p.d.f.). From equation 5, we can see that the estimation of the full posterior at

time t is done by first applying the time evolution model to the previous estimation of the full posterior, leading to what we can call the *belief* at time t, which is in turn multiplied successively by the likelihoods obtained from the different feature spaces in order to get the actual estimation for the posterior.

Note that in our framework, the sequence of words $(z^n)^t$ evolve in time with the acquisition of new images, diverging from the classical Bayesian framework where such sequences would be fixed. Moreover, in spite of the incremental evolution of the dictionary, the representation of each past image is fixed and does not need to be updated.

B. Transition from $t-1$ to t

Between $t-1$ and t, the full posterior is updated according to the time evolution model of the p.d.f., $p\big(S_t|S_{t-1}=j\big)$, which gives the probability of transition from one state j at time $t-1$ to every possible state at time t. It therefore plays a key role in reducing transient detection errors by ensuring the temporal coherency of the detection. Depending on the respective values of S_t and S_{t-1}, this probability takes one of the following values:

- $p\big(S_t=-1|S_{t-1}=-1\big)=0.9$, the probability that no loop-closure event will occur at time t is high given that none occurred at time $t-1$.
- $p\big(S_t=i|S_{t-1}=-1\big)=\frac{0.1}{(t-p)+1}$ with $i\in[0;t-p]$, the probability of a loop-closure event at time t is low given that none occurred at time $t-1$.
- $p\big(S_t=-1|S_{t-1}=j\big)=0.1$ with $j\in[0;t-p]$, the probability of the event "no loop-closure at time t" is low given that a loop-closure occurred at time $t-1$.
- $p\big(S_t=i|S_{t-1}=j\big)$, with $i,j\in[0;t-p]$, is a Gaussian on the distance in time between i and j whose sigma value is chosen so that it is non zero for exactly 4 neighbors (i.e. $i=j-2\ldots j+2$). The size of this neighborhood is adjusted depending on the frame rate and on the velocity of camera motion. This corresponds to a diffusion of the posterior in order to account for the similarities between neighboring images.

Note that in order to have $p\big(S_t>=-1|S_{t-1}=j\big)=1$ when $j\in[0;t-p]$, the coefficients of the Gaussian used in the last case have to sum to 0.9.

C. Likelihood in a Voting Scheme

In section IV-A, we saw how using multiple feature spaces gave the opportunity to represent an image in different ways. From a perceptual point a view, each representation brings its own piece of information about the state of the world, independently from other feature spaces. This entails computing a likelihood measure for the loop-closure hypotheses S_t for each of the feature spaces considered. From the computational point of view, all the representations rely on the bag-of-words paradigm, providing a generic interface to compute and manage image representations. Therefore, the details given here about the estimation of the likelihood associated to a specific feature space k apply identically to each other feature space.

During the computation of the likelihood associated to the feature space k, we wish to avoid an exhaustive image-to-image comparison of the visual features, as implemented in most of the voting and bag-of-words methods cited in section II. In order to efficiently find the most likely past image I_i that closes the loop with the current one, we take advantage of the *inverted index* associated with the dictionary. The inverted index lists the images from which each word has been seen in the past. Then, during the quantization of the current image I_t with the words $(z_k)_t$ it contains, each time a word is found, we retrieve from the inverted index the list of the past images in which it has been previously seen. This list is used to update the score (originally set to 0) that is assigned to every loop-closure hypothesis $S_t=i$ in a simple voting scheme: when we find a word that has been seen in image I_i, statistics about the word are added to the score (see figure 2). The chosen statistics are inspired from the *term frequency–inverted document frequency (tf–idf)* weighting [28]:

$$\text{tf–idf}=\frac{n_{wi}}{n_i}\log\frac{N}{n_w} \tag{6}$$

where n_{wi} is the number of occurrences of word w in I_i, n_i is the total number of words in I_i, n_w is the number of images containing word w, and N is the total number of images seen so far. From equation 6, we can see that the tf–idf coefficient is the product of the term frequency (i.e. the frequency of a word in an image), by the inverted document frequency (i.e. the inverse frequency of the images containing this word). It is calculated each time a likelihood score is computed, giving increased emphasis to words seen frequently in a small number of images, and penalizing common words (i.e. words that are seen everywhere), according to the most recent statistics.

To summarize, when a word is found in the current image, the images where this word has been previously seen have their scores updated with the tf–idf coefficient associated with the pair $\{$*word–image*$\}$. The score associated with each loop-closure hypothesis $S_t=i$ will be used to compute the corresponding likelihood, as we shall see later on. But before, we must give some details about the computation of the score associated to the event "no loop-closure occurred at time t". Indeed, it is evaluated here as the event "a loop-closure is found with I_{-1}". I_{-1} is a virtual image built at each likelihood computation step with the m most frequently seen words of $(Z_k)_t$ (m being the average number of words found per image): it is the "most likely" image.

The idea is that the score associated with I_{-1} will change depending on the location of the current image, so as to behave as the score of the "no loop-closure" event. When no loop-closure occurs, I_t will be statistically more similar to I_{-1} than to any other I_i, because I_t will have more words in common with I_{-1} than with any other I_i. On the other hand, in a real unambiguous loop-closure situation, the score of I_{-1} will be low compared to the score of the loop-closing image I_i: as the words responsible for this detection are only present in two images (i.e. I_t and I_i), they are not frequently seen words and they are in consequence unlikely to be found in I_{-1}. The design of the virtual image proposed here is also relevant in case of perceptual aliasing (i.e. when I_t comes from a location

that is similar to several previously visited places). In such situation, as multiple past images have equivalent likelihoods, it is important to ensure that I_{-1} receives a score that is in the same order of magnitude as the score of these images, so as to prevent an erroneous loop-closure detection. Here, as part of the most common words composing I_{-1} will originate from the images that are responsible for perceptual aliasing, it is guaranteed that I_{-1} will be granted with an important score (but not necessarily the highest one).

The construction of a virtual image with existing words is similar to the addition of new locations from words sampling used in [23]. In our filtering scheme, the existence of the virtual image can be simulated simply by adding a I_{-1} entry to the inverted index for each of the most frequently seen words. Therefore, if one of them is found in I_t, it will vote for I_{-1} as shown in figure 2 and the corresponding score will be computed as for the "true" images.

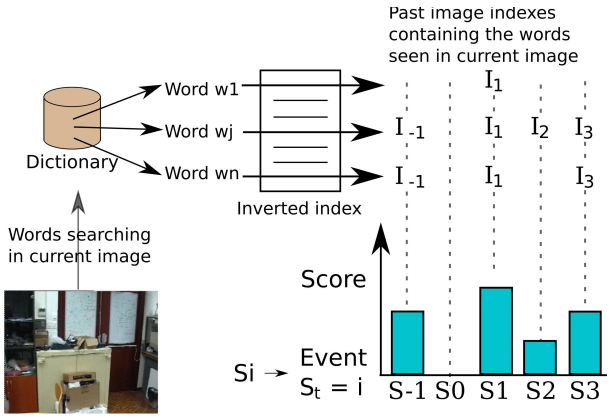

Fig. 2. The voting scheme: the list of the past images in which current words $(z_k)_t$ have been seen is obtained from the inverted index and is used to update the hypotheses' scores.

Once all the words found in the current image have been processed and the computation of the scores is complete, we select the subset $(H_k)_t \subseteq I^{t-p}$ of images for which the *particular* coefficient of variation (c.o.v.) (i.e. particular deviation from the mean of the scores normalized by the mean) is higher than the *standard* c.o.v. (i.e. standard deviation normalized by the mean). $(H_k)_t \subseteq I^{t-p}$ is the subset of the most likely images according to the feature space k. Then, if I_i appears in $(H_k)_t$, the belief at time t (see equation 5) is multiplied by the difference between the particular c.o.v. of I_i and the standard c.o.v., plus 1 (which can be simplified into the difference between the score s_i of the hypothesis and the standard deviation σ, normalized by the mean μ):

$$\mathcal{L}(S_t = i|(z_k)_t) = \begin{cases} \frac{s_i-\mu}{\mu} - \frac{\sigma}{\mu} + 1 = \frac{s_i-\sigma}{\mu} & \text{if } s_i \geq \mu + \sigma \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

The update of the belief for the restricted set of the most likely hypotheses is illustrated in figure 3. The selection done on the hypotheses at this stage makes it possible to simplify the update of the posterior (as only a restricted set of hypotheses is updated), considering that non-selected hypotheses have a likelihood of 1 and therefore multiply the posterior by 1. When all the images of $(H_k)_t$ have been processed for all the feature spaces, the full posterior is normalized.

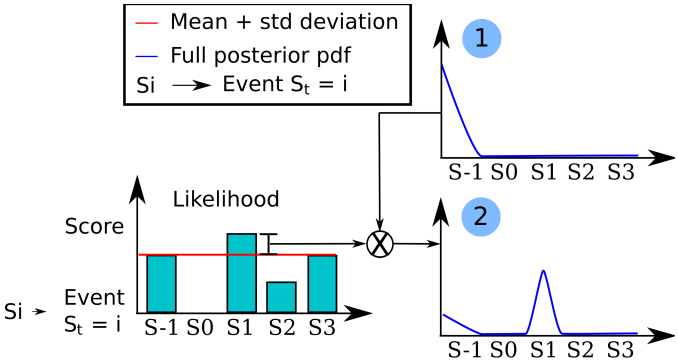

Fig. 3. The belief at time t (frame "1", see equation 5, section IV-A), is updated according to the likelihood model (frame "2"): when the score of a hypothesis is above the mean + standard deviation threshold, the corresponding probability is updated.

D. A Posteriori Hypotheses Management

When the full posterior has been updated and normalized, we search for the hypothesis $S_t = i$ whose a posteriori probability is above some threshold (0.8 in our experiments). However, the posterior does not necessarily exhibit a strong single peak for a unique hypothesis even if a loop-closure occurred. It may rather be diffused over a set of neighboring hypotheses (except for $S_t = -1$). This is mainly imputable to the similarities among neighboring images in time: some of the words commonly found in I_t and I_i are also probably in I_{i-1} or I_{i+1} for example. Thus, instead of searching for single peaks among the full posterior, we look for a hypothesis for which the sum of the probabilities over neighboring hypotheses is above the threshold (the neighborhood chosen here is the same as the neighborhood selected for the diffusion in section IV-B).

When a hypothesis fulfills the above condition, a multiple-view geometry algorithm [11] helps discarding outliers by verifying that the two images of the loop-closure (i.e. I_t and I_i) satisfy the epipolar geometry constraint, which would imply that they share some common structure and that they could hence come from the same 3D scene. To this end, a RANSAC procedure entails rapidly computing several camera transformations by matching SIFT features between the two frames, discarding inconsistent ones using a threshold on the average reprojection error. If successful, the algorithm returns the 3D transformation between x_t and x_i (i.e. the viewpoints associated with I_t and I_i) and the hypothesis is accepted. Otherwise, the hypothesis is discarded. However, even if a hypothesis has been discarded by the multiple-view geometry algorithm, its a posteriori probability will not fall to 0 immediately: it will diffuse over neighboring images during the propagation of the full posterior from t to $t+1$. Thus, correct hypotheses erroneously discarded by epipolar geometry will be reinforced by the likelihoods of further time instants until a valid 3D transformation is found. Note that since SIFT features are extracted from the images and stored

during the online dictionary construction, we do not need to process the images again when applying the multiple-view geometry algorithm.

V. Experimental Results

We obtained results[1] from several indoor and outdoor image sequences grabbed with a single monocular handheld camera (i.e. a simple camcorder with a 60° field of view and automatic exposure). In this paper, we present the results obtained from two experiments: an indoor image sequence with strong perceptual aliasing and a long outdoor image sequence. In both experiments, illumination conditions remained constant: the indoor sequence has been captured under artificial lighting conditions, while the length of the outdoor one (i.e. nearly 20 minutes) was too short to experience changes in lighting conditions.

A. Indoor experiment

The overall camera trajectory followed during this experiment is shown in figure 4 using three different styles. When the posterior is below the threshold, the trajectory is shown with a blue (dotted) line. When it is above the threshold and the epipolar constraint is satisfied, a loop-closure is detected and the trajectory is shown with a green (dashed) line. But, when the posterior is above the threshold and the epipolar constraint is not satisfied, the loop-closure hypothesis is rejected and the trajectory is shown with a red (circled) line.

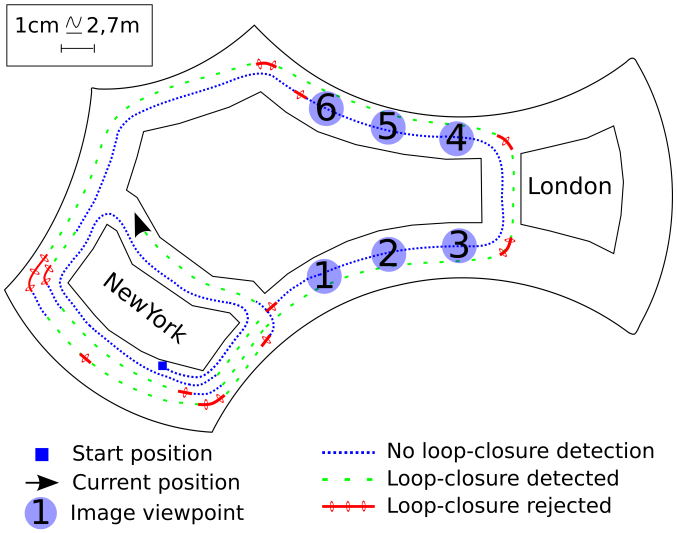

Fig. 4. Overall camera trajectory for the indoor image sequence. A first short loop is done around the "New York" elevators on the left before going to the "London" elevators on the right. The short loop is travelled again when the camera is back from the "London" elevators following the top-most corridor on the plan. Then, the camera repeats the long loop (i.e. to the "London" elevators and back) before ending in front of the "New York" elevators. The numbers in the circles indicate the positions from which the images shown in figure 5 were taken. See text for details about the trajectory.

As we can see in figure 4, the trajectory is shown with a blue (dotted) line every time the camera is discovering unexplored areas, in spite of the strong perceptual aliasing present in the corridors to and from the "London" elevators (see figure 5 for examples of the images composing the sequence). During the run, no *false positive* detections were made (i.e. when a loop-closure is detected whereas none occurred), thus demonstrating the robustness of our solution to perceptual aliasing.

Fig. 5. Top-most corridor (top row) and bottom-most corridor (bottom row) image examples, showing the high level of perceptual aliasing in the environment. The numbers in the circles help associating the images with the positions labelled in figure 4.

From figure 4, we can also see that the trajectory is shown with a green (dashed) line most of the time spent in previously visited places, indicating that *true positive* detections were made (i.e. when a loop-closure occurs and it is correctly detected). Figure 6 gives an example of a true positive detection.

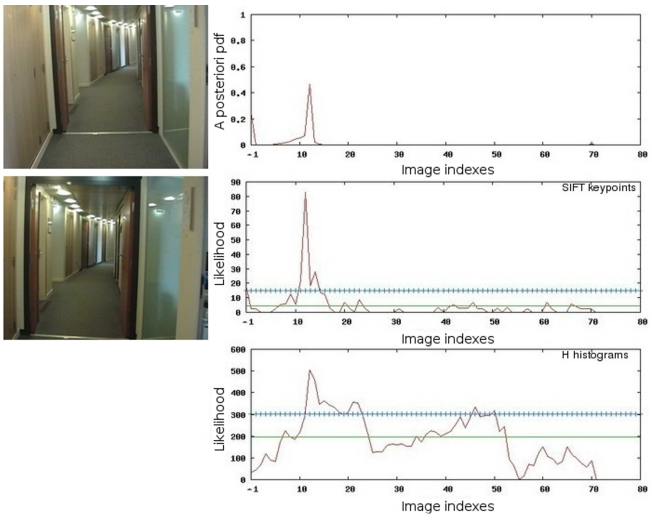

Fig. 6. First loop-closure detection for the indoor image sequence. The full posterior and the likelihood computed from the SIFT and H histograms feature spaces are shown, along with the current image I_t (top left) and the loop-closing image I_i (bottom left). Likelihoods are obtained from the scores (tf–idf) of the different hypotheses. Also shown with the likelihoods are the score mean (solid green) and the score mean + standard deviation threshold (blue crosses). As it can be seen, the likelihood is very strong around images corresponding to hypotheses 10 to 13, causing the sum of the corresponding probabilities in the posterior to reach the 0.8 threshold. Also, it clearly appears here that I_t and I_i come from very close viewpoints.

During passings in already explored places, it may be noticed that the line representing the trajectory switches from green (dashed) to red (circled) each time the camera was

[1]Videos available at http://animatlab.lip6.fr/AngeliVideosEn, but also at http://ieeexplore.ieee.org as supplemental material to this paper.

turning around corners. In these particular cases, loop-closure detection fails only because the epipolar constraint is not satisfied: the a posteriori probability of loop-closure is above the threshold but, due to the large and fast rotations made by the camera, precise keypoints associations are difficult. Indeed, in this narrow indoor environment, when the camera is turning around corners, the viewpoint variation between current and loop-closing images may be large, resulting in small overlap between these images and preventing SIFT features from matching correctly. This corresponds to *false negative* detections (i.e. when a loop-closure occurs but it is not detected).

When considering the trajectory of the camera with more attention, it may be observed that the first loop-closure detection that should be done (i.e. when the camera reaches again its starting position for the first time, during its first travel behind the "New York" elevators) is missed and the trajectory remains shown with a blue (dotted) line. This is imputable to the low responsiveness of the probabilistic framework: the likelihood associated with a particular hypothesis has to be very high relative to the other likelihoods to trigger a fast loop-closure detection. Usually, the likelihood associated with a hypothesis must have a good support during 2 or 3 consecutive images in order to trigger a loop-closure detection. The responsiveness of our system is governed by the transition model of the probabilistic framework: we assume that the probability of remaining in a "no loop-closure" event is high (i.e. 0.9, see section IV-B). Decreasing this probability to lower values makes it possible to detect loop-closures faster (i.e. with fewer images required), but this also produces false positive detections, which is not acceptable. The delay involved here therefore enhances the robustness to transient detection errors, considering only hypotheses with repeated support over time as possible candidates for loop-closure.

During the run, there was only one case where the probability was above the threshold but the selected hypothesis was wrong and it has been conveniently rejected by the multiple-view geometry algorithm. This event, that can be considered as a *false alarm*, can be identified in figure 4 as the red (circled) portion of the trajectory that occurs when the camera is coming back for the first time from the "London" elevators (just near the 6^{th} circle on the figure). This false alarm can be explained by the strong perceptual aliasing that makes the corridors to and from the "London" elevators look the same (see figure 7): since our bag-of-words algorithm relies on the occurrence of the words rather than on their position, the current image may look like a past image but the structures of the scenes may not be consistent, thus preventing the epipolar constraint from being satisfied.

In order to test the robustness of the detection to camera viewpoint changes, we rotated the camera around its optical axis when passing behind the "New York" elevators for the second and third times. As shown by the green (dashed) line representing the trajectory during these periods, the loop-closure detection results were not affected. The figure 8 gives an example of loop-closure detection with different camera orientations between current and loop-closing images. The loop-closure detection shown in this figure corresponds to the third passing of the camera behind the "New York" elevators. This is why we observe two distinct peaks on the likelihoods: two hypotheses are valid in this case, because I_t closes the loop with images from the first and the second visits. But due to the temporal coherency of the p.d.f., the hypotheses that have high a posteriori probabilities are those from the second passing.

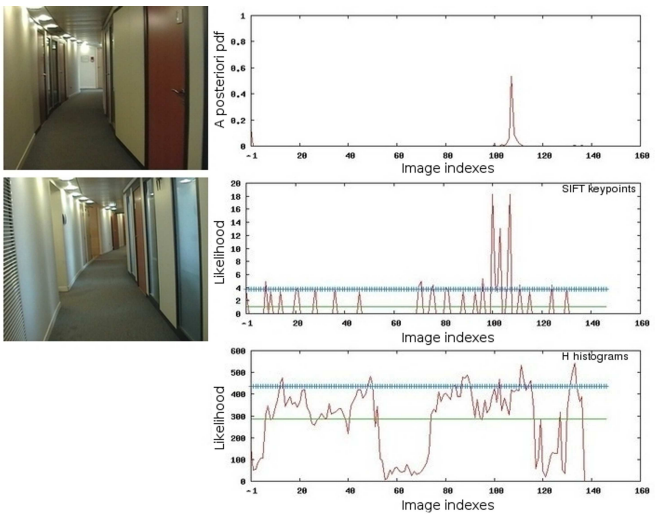

Fig. 7. The only false alarm due to perceptual aliasing: as we can see, the likelihoods are confused (we can note two similar high peaks on the SIFT's likelihood, while the H histograms' likelihood does not give helpful information) and the images look very similar. This hypothesis has been rejected by the multiple-view geometry algorithm.

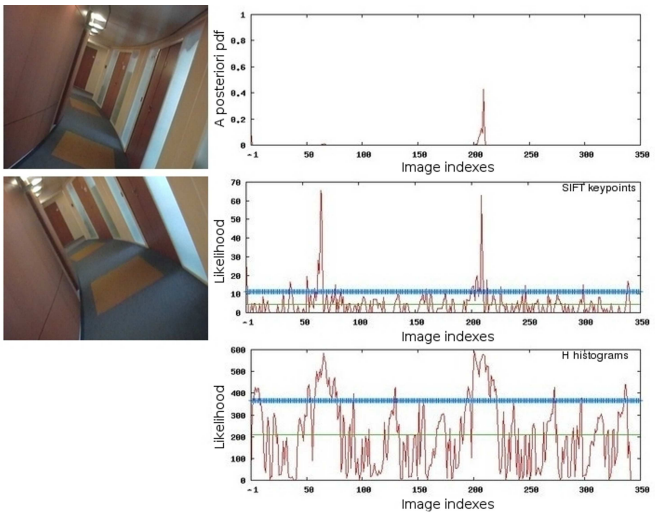

Fig. 8. Another loop-closure detection for the indoor image sequence. Although there is a significant camera viewpoint difference between current and past images, the loop-closure is correctly detected.

B. Outdoor experiment

During this second experiment, images were taken outdoor with a handheld camera while turning around the laboratory's building (figure 9 gives examples of images from this sequence).

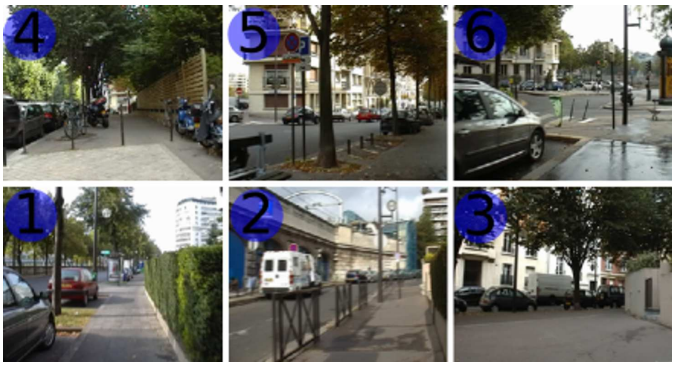

Fig. 9. Examples of the images composing the outdoor sequence. The numbers in the circles help associating the images with the positions labelled in the figure 10.

The overall camera trajectory followed during this experiment is shown in figure 10 using the same style conventions as before. Here, we introduced red-green (circled-dashed) lines to denote fast alternations of true positive and false negative detections that occur when people or cars are passing in front of the camera, causing correct hypotheses to be rejected because not enough point correspondences can be found to satisfy the epipolar geometry constraint. These events (of which one example is given in figure 11) demonstrate the robustness of the probabilistic framework to transient detection errors: even though images are occluded by people or cars, correct loop-closure hypotheses are selected (i.e. they have a high a posteriori probability), but since the epipolar constraint cannot be satisfied, they cannot be fully validated to be accepted as true positive loop-closure detections.

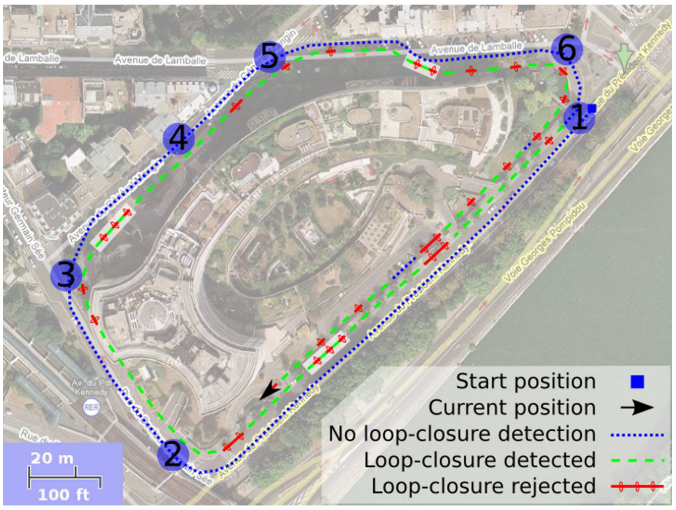

Fig. 10. Overall camera trajectory for the outdoor image sequence. Two loops are done around the "Lip6" laboratory, starting near the top-right end of the building on the image (indicated by the square) and ending at its bottom-left end. The path in front of the building (i.e. running parallel to the river) is thus travelled three times. The style conventions for the trajectory are the same as in figure 4, with the introduction here of red-green (circled-dashed) lines to denote fast alternations of true positive and false negative detections. Red-green (circled-dashed) lines are painted over white rectangles to distinguish them easily. See text for details about the trajectory.

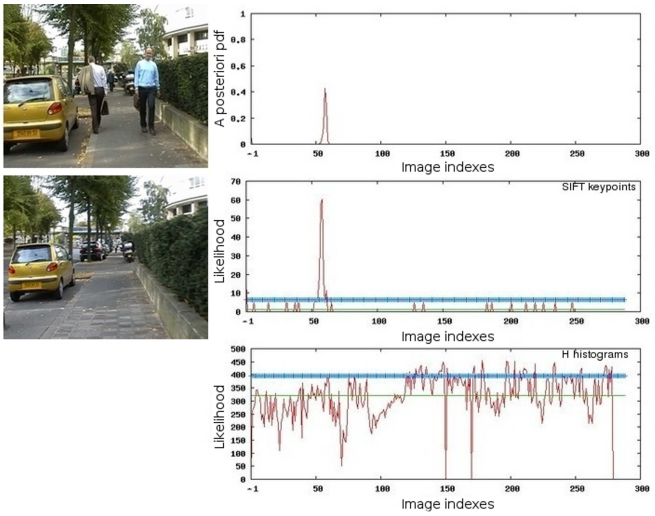

Fig. 11. Robustness of the probabilistic framework to transient detection errors: although current image is partially occluded by pedestrians, a correct loop-closure hypothesis is selected, but it is rejected by the multiple view geometry algorithm.

As in the indoor experiment, no false positive detections were made, whereas multiple true positives were found (see figure 12). Also, we can see from figure 10 that the first loop-closure detections occur tardily when the camera is coming back to its starting position, revealing again the low responsiveness of the probabilistic framework.

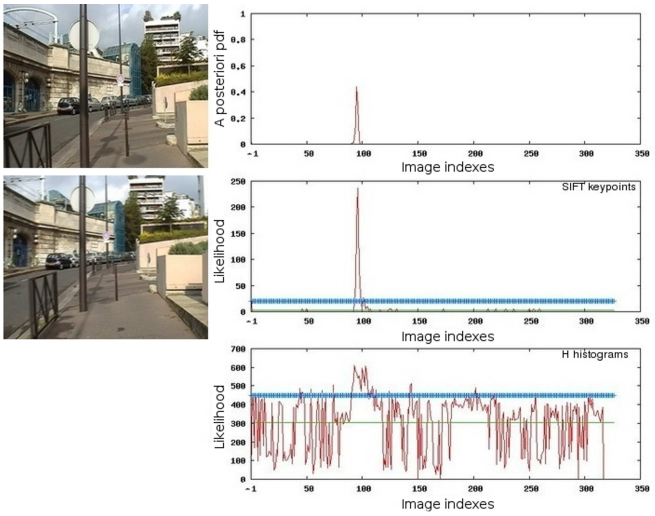

Fig. 12. Example of a true positive loop-closure detection for the outdoor image sequence. Again, we can observe that the likelihood from the SIFT feature space is very high and discriminative.

C. Influence of the visual dictionaries

In this section we will study the influence of the different visual dictionaries used here (i.e. SIFT features and H histograms) for loop-closure detection. To this end, we tried to perform loop-closure detection using only either SIFT features or H histograms. Although those tests have been done using both image sequences, the indoor one produces more valuable results since more loop-closures are done during the travel of

the camera and because the indoor environment is much more diversified.

H histograms only carry colorimetric information, without any shape nor structure information. Therefore, the corresponding likelihood is always confused, and it will never be very peaked over one particular hypothesis unless the corresponding image contains specific colors that are seen nowhere else. However, H histograms can help distinguishing similarly structured environments that only differ in their colors (e.g. two corridors having the same dimensions but whose walls are painted with different colors). When used alone, H histograms cannot trigger a loop-closure detection. But when used in combination with SIFT features, they enhance loop-closure detection, improving notably the overall responsiveness of the probabilistic framework. Indeed, as shown in figure 13, we can see that the posterior obtained when using both SIFT features and H histograms is higher than when using SIFT features only. This is because H histograms' likelihood, although not discriminative enough to trigger a loop-closure detection, is a higher around the loop-closing hypothesis, and so it reinforces the votes from the SIFT feature space when updating the posterior.

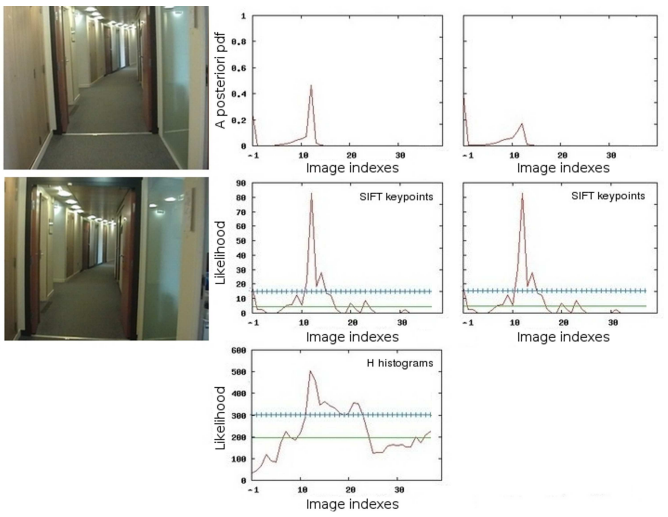

Fig. 13. Loop-closure detection enhancement using color and shape information in the indoor image sequence: when H histograms are combined to SIFT features (left part), the a posteriori probability is higher than when using SIFT features alone (right part).

Using SIFT features in conjunction with H histograms therefore enhances the responsiveness of the algorithm, making it able to detect loop-closures sooner, especially when the camera is back to its starting position for the first time: loop-closures are detected 2 or 3 images before when both feature spaces are involved. Table I gives additional clues for this improvement, with information about the loop-closure detection performances for the indoor and outdoor image sequences when using SIFT features alone or in conjunction with H histograms. Given are the number of images composing each sequence ("#img"), the corresponding number of loop-closures ("#LC", determined at hand from the camera trajectory), the rate of true positive detections ("%TP", the percentage of loop-closures correctly detected), and the number of false alarms ("#FA", erroneous hypotheses that receive a high probability but that are rejected by the multiple-view geometry algorithm).

TABLE I
COLOR INFORMATION IMPROVEMENTS

Sequence	#img	#LC	%TP	#FA
Indoor SIFT + H	388	217	80	1
Indoor SIFT	388	217	68	0
Outdoor SIFT + H	531	301	71	0
Outdoor SIFT	531	301	70	0

From table I, we can see that when adding color information, the true positive rate is improved: this is notably remarkable in the indoor sequence where the increase in recognition performances is 12%. On the outdoor sequence on the other hand, improvements are less significant. This is due to the impressive reliability of the SIFT features in this sequence. Indeed, as SIFT features are robust to scale variations in the images, the important depth of the outdoor scenes enables long term recognition of these features along the trajectory of the camera. Hence, adding color information in this case does not dramatically improve the number of correct loop-closure detections. We can also see in table I that adding color information has the unwanted effect of producing more false alarms: when using SIFT features only, no false alarms were raised for the indoor image sequence, whereas one was when combining them with H histograms (see section V-A).

D. Performances

During the experiments, the dictionaries were built online in an incremental fashion from images of size 240x192 pixels, enabling real-time performances with a Pentium Core2 Duo 2.33GHz laptop in both indoor and outdoor experiments.

Table II gives the length of the different sequences tested (with corresponding number of images), the CPU time needed to process them, and the sizes of the different dictionaries at the end of the run (expressed in number of words). For both sequences (i.e. indoor and outdoor), we give the performances obtained when SIFT features are used alone or in combination with H histograms.

TABLE II
PERFORMANCES

Sequence	Length	#img	CPU	#SIFT	#H hist.
Indoor SIFT + H	6m28s	388	2m52s	9201	7284
Indoor SIFT	6m28s	388	1m33s	9201	0
Outdoor SIFT + H	17m42s	531	10m16s	39175	18408
Outdoor SIFT	17m42s	531	6m48s	39175	0

For the indoor experiment, images were grabbed at 1Hz: the camera was moved along medium sized corridors, with curved shape and suddenly appearing corners, motivating the choice for a reasonable framerate in order for consecutive images to share some similarities. For the outdoor experiment however, images were grabbed with a lower framerate (i.e. 0.5Hz): outdoor images grabbed at distant time instants share some similarities because of the high depth of outdoor scenes.

From table II, we logically observe that when using SIFT features only, the CPU time needed to process a sequence is significantly lower than when H histograms are involved too: the overall processing is about 40% faster in the first case. However, with both feature spaces enabled, real-time processing is still achieved and, as mentioned before, the responsiveness of the probabilistic framework is enhanced, without causing false positive detections to appear. When processing an image, the most time consuming step is feature extraction and matching with the words of the corresponding dictionary. When trying to match a feature with the visual words of the dictionary, the search is done with logarithmic-time complexity in the number of words due to the tree structure of the dictionary [26]: real-time performances could not have been obtained with linear-time complexity in the number of words in view of the dictionary sizes involved here.

For the outdoor experiment, the overall camera trajectory was about 1.3km and a bit less than 40000 words were created (when considering the SIFT case only) from 531 images. In the results obtained by the authors of [23], the data collection for dictionary construction has been done over 30km, using 3000 images and generating approximately 35000 words. It is obvious that our model needs far more words than the solution proposed in [23], and the intuitive explanation of this is twofold. First, in our online dictionary construction, we cannot afford data rearranging, which would make it possible to obtain a more compact representation. Secondly, in order for the tf–idf weighting used here to perform efficiently, discriminative words are preferable in order to select unambiguous hypotheses. As shown in [10], the size of the cluster representing the words has a direct influence on the word's distinctiveness: a higher distinctiveness is obtained with a smaller cluster size, i.e. a larger dictionary size. The parameters used here are found experimentally to perform well on all the encountered environments.

VI. Discussion and Future Work

The solution proposed in this paper is a completely incremental and online vision-based method allowing loop-closure detection in real-time. The bag-of-words framework introduced in [10] and used here provides a simple way to manage multiple image representations, taking advantage of information gathered from distinct heterogeneous feature spaces. Moreover, building the dictionaries in an incremental fashion entails "learning" only that part of the environment in which the robot is operating, while bag-of-words methods applied to robotics usually use a static dictionary (e.g. [20], [21], [23]) learned beforehand from a training data set supposed to be a good representation of the environment. The consequence is that our system is able to work indoor and outdoor without hand-tuning the dictionary, and without prior information on the environment type.

The results presented here show the robustness of our solution to perceptual aliasing. However, the more complex probabilistic framework described in [23] handles it more properly, taking it into account at the word level (i.e., the input information level) while, in our case, it is managed at the detection level (i.e., the output level), when hypotheses are checked by the epipolar geometry algorithm. Still, the evaluation of the distinctiveness of every word proposed in [23] cannot be done incrementally because, to evaluate the co-occurrences of the words, representative images of the entire environment have to be processed beforehand. In our method, the distinctiveness of the words is taken into account using the online calculated tf–idf coefficient: the words seen multiple times in the same image will vote with a high score for this image (i.e. high tf), while the words seen in every images will have a small contribution (i.e. low idf).

The probabilistic framework presented here poorly handles the management of loop-closure hypotheses. Indeed, a new entry is added to the posterior each time a new image is acquired, while the evaluation of the corresponding hypotheses (i.e. checking if whether or not the newly acquired image closes the loop with one of the past images) is done afterwards: in other words, a new image is added to the model irrespectively of the loop-closure detection results. In future work, a topological map of the environment could be directly created by adding only images that do not close a loop with already memorized ones. These events would therefore represent positions in the environment, linked by their proximity in time and space, and not only images linked sequentially in time. This would avoid the presence of multiple high peaks due to the co-existence of multiple images taken from the same position (see figure 8).

In future work, we will adapt our approach to a purely vision-based SLAM system like [6] so as to reinitialize the SLAM algorithm when the camera position is lost or when there is a need to self-localize in a map acquired beforehand. The metrical information about the camera's pose coming from SLAM could help improving the definition of a location's neighborhood, using spatial transitions between adjacent locations instead of time indexes. As mentioned above, this would make it possible to fuse images taken from close metric locations to build a topological map of the environment.

Finally, other feature spaces could be explored, implementing for instance one of the visual descriptors tested in [25], whereas relative spatial positions between the visual words could be used to improve matching. Loop-closure detection at different moments of the day should also be experienced, so as to test the robustness of our solution to varying lighting conditions.

VII. Conclusion

In this paper, we have presented a fast and incremental bag-of-words method for performing loop-closure detection in real-time, with no false positive detections on the obtained experimental results even under strong perceptual aliasing conditions. We demonstrated the quality of our approach with results obtained in indoor and outdoor environments, reaching real-time performances even in long image sequences. Our approach calls upon a Bayesian filtering framework with likelihood computation in a simple voting scheme and should be extended to SLAM reinitialization in a near future.

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the reviewers for their useful comments on reviewing the paper.

REFERENCES

[1] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots - I. a review of localisation strategies," *Journal of Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.

[2] J.-A. Meyer and D. Filliat, "Map-based navigation in mobile robots - II. a review of map-learning and path-planing strategies," *Journal of Cognitive Systems Research*, vol. 4, no. 4, pp. 283–317, 2003.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i," *IEEE Robotics and Automation Magazine*, vol. 13, no. 1, pp. 99–110, 2006.

[4] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (slam): Part ii," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[5] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "2d simultaneous localization and mapping for micro aerial vehicles," in *European Micro Aerial Vehicles (EMAV)*, 2006.

[6] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[7] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, February 2007.

[8] M. E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *IEEE conference on Computer Vision and Pattern Recognition*, 2006.

[9] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV04 workshop on Statistical Learning in Computer Vision*, 2004, pp. 59–74.

[10] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *IEEE International Conference on Robotics and Automation*, 2007.

[11] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, 2004.

[12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *IEEE International Conference on Robotics and Automation*, May 1999.

[13] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image retrieval system with monte carlo localization," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 208–216, 2005.

[14] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, Mexico: IJCAI, 2003.

[15] M. Pupilli and A. Calway, "Real-time visual slam with resilience to erratic motion," in *IEEE Computer Vision and Pattern Recognition*, 2006.

[16] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardòs, "Mapping large loops with a single hand-held camera," in *Proceedings of Robotics: Science and Systems*, 2007.

[17] B. Williams, P. Smith, and I. Reid, "Automatic relocalisation for a single-camera simultaneous localisation and mapping system," in *IEEE Iternational Conference on Robotics and Automation (ICRA)*, 2007.

[18] J. Kosecká, F. Li, and X. Yang, "Global localization and relative positioning based on scale-invariant keypoints," *Robotics and Autonomous Systems*, vol. 52, pp. 209–228, 2005.

[19] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *IEEE International Conference on Robotics and Automation*, 2000.

[20] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-invariant features," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 36, no. 2, pp. 413–422, April 2006.

[21] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[22] K. L. Ho and P. Newman, "Detecting loop closure with scene sequences," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 261–286, 2007.

[23] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, 2007.

[24] D. Lowe, "Distinctive image feature from scale-invariant keypoint," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[25] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, June 2003, pp. 257–263.

[26] D. Filliat, "Interactive learning of visual topological navigation," in *To appear in the proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008.

[27] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 246–253.

[28] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, 2003.

Adrien Angeli received in 2005 the Master's degree in computer engineering from the Ecole Centrale d'Electronique, Paris, France and the Master's degree in artificial intelligence from the Université Pierre et Marie Curie, Paris, France. He is currently a Ph.D. student with the Université Pierre et Marie Curie. His research interests include vision-based localization and SLAM applications to robotics.

David Filliat graduated from the Ecole Polytechnique in 1997 and obtained a PhD in robotics from Paris VI university in 2001. After 3 years of work for the robotic programs in the French armament procurement agency, he is now assistant professor at Ecole Nationale Supérieure de Techniques Avancées. His main research interest are perception, navigation and learning in the frame of the developmental approach to autonomous mobile robotics.

Stéphane Doncieux is Assistant Professor at Paris 6 University and is responsible for the SIMA (Integrated, Mobile and Autonomous Systems) research team of ISIR. He has been trained as an engineer and holds a PhD in computer science. He works on the autonomous design of control architectures thanks to evolutionary algorithms and on adding decisional autonomy to flying robots. He is also heading the ROBUR project of ISIR, which aims at building an autonomous flapping-wing robot.

Jean-Arcady Meyer is Emeritus Research Director at CNRS. He trained as an engineer, he graduated in Human and Animal Psychology, and received a PhD in Biology. He is the founder of the journal Adaptive Behavior, a former Director of the International Society for Adaptive Behavior and a current Director of The International Society for Artificial Life. His primary scientific interests are in adaptive behaviors in natural and artificial systems. He is the main coordinator of the Psikharpax project.

Incremental topo-metric SLAM using vision and robot odometry

Stéphane Bazeille & David Filliat
ENSTA ParisTech, Unité Electronique et
Informatique, 32 Boulevard Victor, 75015 Paris,
FRANCE.
stephane.bazeille@ensta-paristech.fr,
david.filliat@ensta-paristech.fr

***Abstract*— We address the problem of simultaneous localization and mapping by combining visual loop-closure detection with metrical information given by the robot odometry. The proposed algorithm builds in real-time topo-metric maps of an unknown environment, with a monocular or omnidirectional camera and odometry gathered by motors encoders. A dedicated improved version of our previous work on purely appearance-based loop-closure detection [1] is used to extract potential loop-closure locations. Potential locations are then verified and classified using a new validation stage. The main contributions we bring are the generalization of the validation method for the use of monocular and omnidirectional camera with the removal of the camera calibration stage, the inclusion of an odometry-based evolution model in the Bayesian filter which improves accuracy and responsiveness, and the addition of a consistent metric position estimation. This new SLAM method does not require any calibration or learning stage (i.e. no *a priori* information about environment). It is therefore fully incremental and generates maps usable for global localization and planned navigation. This algorithm is moreover well suited for remote processing and can be used on toy robots with very small computational power.**

***Keywords*: SLAM, uncalibrated camera, robot odometry, hybrid topo-metric map.**

I. INTRODUCTION

To navigate in an unknown environment a robot requires the ability to build a map and to localize itself using a process named Simultaneous Localization And Mapping (SLAM) [29]. The field of SLAM can be broadly divided into topological and metrical approaches. The topological approach models the environment as a graph of discrete locations and often leads to simple solutions [11], [3]. It is often an easy to build map, suitable for many kinds of environment and for human interactions. Its main drawback is the lack of geometric information that only allows localization in previously mapped areas and local navigation with non optimal path planning. On the contrary, the metrical map is explicitly based on measured distances and positions [5]. The representation of the environment is geometric and clearly corresponds to the real world. The localization can be done continuously and planned navigation may be more precise. The major problem is to ensure geometry consistency between position and perceptions which makes the map hard to build. Number of approaches have attempted to capitalize on the advantages of the two representations (e.g., [19]). For instance, local metrical maps can be embedded into graphs to enhance scalability [8]. Other graph-based solutions can be used to infer a precise metrical position for the robot, while still allowing for large scale mapping [17].

Without any prior information about the environment, and only using a monocular calibrated camera we have demonstrated that real-time topological SLAM is possible [1]. This method presents many advantages such as its simplicity, its speed, the lack of learning stage, and its efficiency (low false alarm rate), but the lack of metrical information makes the map ill-posed for navigation. In this article, we present an improved version of this method including the metrical information given by the robot odometry. Our new framework is calibration-free, incremental and real-time and allows to build hybrid topological-metrical maps usable for robot guidance. The odometric information is easy to acquire because often provided on robots. It also greatly complements the image data because it remains available notably in case of vision system failure (e.g. device problem, sensor occlusion, strong lighting change, dark areas).

In Section 2, we present related work on visual loop-closure detection, topological mapping and hybrid topo-metric mapping. In Section 3, we recall our previous work on the visual topological SLAM and we present our new framework on calibration free topo-metrical SLAM. In Section 4, we show experimental results and we conclude in Section 5 with a discussion and our future work.

II. RELATED WORK

In Simultaneous Localization And Mapping solutions, the traditional range and bearing sensors are now often replaced by camera (e.g. [2], [5]). It provides indeed many advantages such as smaller size, lighter weight, lower energy consumption, and above all a richer environmental information that is usable as the only environment information. Among all the approaches, we are more particularly interested in vision-based topological SLAM methods (e.g., [1], [4], [18]). The main idea of these approaches is to seek for the past images that look similar to the current one and consider they come from close viewpoints. This matching problem is called loop-closure detection. We have demonstrated in [1] a vision-based loop-closure detection method with a single monocular calibrated camera. The method uses Bayesian filtering, a simple voting scheme to estimate loop-closure likelihood and a multiple-view geometry stage to discard outliers.

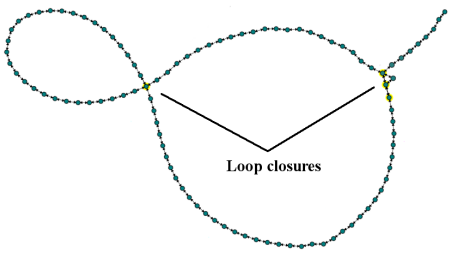

Fig. 1. Topological map (Museum sequence).

Given this framework, our objective is the integration of information to the topological map (Fig. 1) so as to obtain a map with which robot guidance is possible. The most appealing solution to this problem is probably the use of visual odometry, where images coming from neighboring nodes or image sequences taken between nodes are matched to estimate the robot displacement [11], [23], [17], [28]. Instead of estimating node positions, another solution is to use visual servoing, also known as vision-based robot control which uses feedback information extracted from a vision sensor to control the motion of a robot [6]. The robot can then be directly guided to the neighboring nodes without explicitly computing their relative positions. The advantage of these two approaches is to maintain the use of the vision sensor only but they require a lot of processing and are not robust in dark or over-exposed areas for example. So, like several authors [10], [25], we have chosen to use the information given by robot odometry. It adds another sensor but it greatly reduces processing requirement and complements visual sensor in case of perception failure.

Odometry is often used on robots, whether they be legged or wheeled, to estimate their position relative to a starting location. The major problem with robot odometry is a cumulative error due to the integration of noisy measurements that makes the position estimate more and more inaccurate over time. As a consequence, long term use of odometry requires complementary information to enable a correction of this cumulative errors and to produce a consistent topo-metric map. As loop-closure detection provides a reliable information about position we can apply a relaxation algorithm to estimate the position of nodes that best satisfied loop-closure constraints. Several relaxation methods exist to deal with this problem [16], [7], [12]. Also, recent solutions are very interesting and efficient [24], [15] to solve the particular graph-based formulation of SLAM problem in which the poses of the robots are modeled by nodes in a graph, and constraints between poses resulting from odometry are encoded in the edges.

Visual SLAM approaches have used either standard perspective [5],[4] or omnidirectional ([14], [3]) camera as input. Omnidirectional cameras are interesting for SLAM applications because they give a richer information to characterize scenes as a single image can visually describe the position of the robot regardless of its orientation. For example, omnidirectional camera allow loop-closure detections when the robot take a previous path in the opposite direction. Many approaches rely on a particular type of camera with its associated calibration [5],[27], while some approaches can be used with uncalibrated camera [26], [13] which greatly simplifies the application to different robots. Our approach is generic as it is calibration-free and as it can use either perspective or omnidirectional camera as input.

III. Limitations of loop-closure detection for topo-metric mapping

Building globally coherent and locally precise topo-metric maps using robot odometry and graph relaxation requires a robust and reliable loop-closure detection algorithm. In particular, we need a metrically accurate and responsive detection, and the strict absence of false alarm which would lead to globally incoherent maps. Our previous approach on visual loop-closure detection [1] presented some limitations to be used in this framework.

A first limitation is in the definition of loop-closure by itself. In our original approach, as in others [4], loop-closure were validated using multiple-view geometry between the matching images. This policy define loop-closure by the fact that the robot sees the same scene, but does not enforce that the robot positions are very close, and scale ambiguity in multiple-view geometry prevents to recover the real robot displacement. In order to include the loop-closure constraints in a graph relaxation algorithm, we therefore chose to devise a more constrained loop-closure validation stage to only accept loop closure detections with very close robot positions (see section IV-E). While this could appear as a strong constraint, it in fact quite natural for robots in indoor environments as robots often pass very close to the same positions, for example when crossing doors (e.g., Fig. 7).

A second limitation of our approach is the lack of temporal consistency of loop-closure detection. For example on Fig. 1 the loop-closure detected by the original algorithm were between images 49-80, 9-102, 10-104, 9-105, 9-106 while the ground truth were between images 47-78, 48-79, 49-80, 5-102, 6-103, 7-104. While these loop-closure detection are visually correct according to our previous definition, two major problems can be seen on this example: first the detection are not temporally consistent (10-104, 9-105) and also, successive loop-closure on the same node are detected (9-105, 9-106). This is normally impossible because the images are only processed when the robot has moved for a given distance. Including these inconsistencies, the graph relaxation would lead to large errors in the optimized map. This problem has been solved by the inclusion of an odometry-based evolution model (see section IV-D) that bias loop-closure detection to follow the robot trajectory temporal evolution.

Finally, the responsiveness of the algorithm was too low for complex indoor trajectories where loop-closure can only be detected during one or two images. As an example in indoor environments, common trajectories are mostly seen when a door is crossed for a very short distance. In these

cases, the previous approach leads to many missed loop-closure and provides very few constraints for map relaxation and odometry correction. This problem has also been solved through the odometry-based evolution model.

IV. Overview of the approach

In our previous work [1] we have developed a real-time vision-based topological SLAM framework. This method is fully incremental (i.e. the system can be used without any *a priori* information about the environment), and only uses appearance information from a single calibrated camera. The environment model is learned on-line, as the robot discovers its surroundings.

A. The Bayesian filtering using bags of visual words

To solve the image-to-node matching problem based on a similarity measure between the current image and the images of a node previously visited, we choose to use a maximum *a posteriori* scheme which exploits the similarity of image sequences to ensure the temporal consistency and reduce false alarms (e.g. [21]), instead of the more common maximum likelihood which only consider the current image for matching (e.g. [3]). A short overview of the Bayesian filtering framework is provided here for clarity.

The method searches for the node N_i of the map that is the more similar to the current image I_t, in other words, it searches for the node N_i that maximizes the probability of loop-closure with the current image:

$$N_i = argmax_{i=0,...,n} p(S_t = i | I_t, M) \tag{1}$$

where $S_t = i$ is event "I_t comes from N_i" and $M = N_0, ..., N_n$ is the map. Bayes rule, marginalization and Markov assumption [1] lead to the incremental computation of the *a posteriori* probability as follow:

$$p(S_t|I_t, M) = \eta . \underbrace{p(I_t|S_t, M)}_{\text{likelihood model}} . \underbrace{\sum_{j=0}^{n} \underbrace{p(S_t|S_{t-1} = j, M)}_{\text{transition model}} \underbrace{p(S_{t-1} = j|I_{t-1}, M)}_{\textit{a priori} \text{ probability}}}_{\text{prediction}}$$

In this equation, the prediction is computed using the *a priori* probability (i.e. the probability at the previous time step) multiplied by an evolution model $p(S_t|S_{t-1} = j, M)$ diffusing the probability of a node to its neighbors to take into account the robot motion since the last localization. Then, the result of this computation called prediction is multiplied by the likelihood (number of correspondences between images through a voting scheme) to obtain the *a posteriori* probability. The likelihood model is computed using a representation of images as a set of unordered SIFT features [20] taken from a dictionary (i.e. the bags of visual words model [9]). An inverted index makes it possible to very efficiently compute this likelihood in time linear with the number of visual words of the current image. To discard outliers, we use a multiple-view geometry stage as in [18]. The posterior probabilities above a threshold are first sorted and then the essential matrix between the two loop-closure images [22] are computed in the descending order.

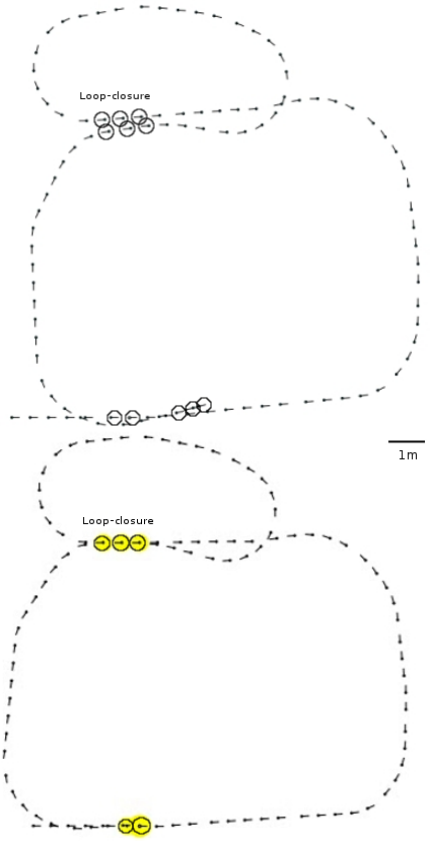

Fig. 2. Topo-metric maps (Museum sequence). (Top) the raw odometry, the loop-closure detections are highlighted with black circles, (Bottom) corrected odometry after relaxation.

B. The addition of robot odometry information

We did four main modifications in order to incorporate the odometry information in the map (a complete processing diagram of the new algorithm is shown Fig. 3):

- images are now acquired with the odometry when the robot has moved for a given distance or turned of a given angle. This image acquisition policy enforces a more regular sampling of positions in the environment, independent of the robot velocity and also reduces the computational burden of the algorithm when the robot is not moving.
- the Gaussian evolution model was not precise enough and has been replaced by an odometry-based evolution model. Through a probabilistic model of odometry, the evolution model can take into account not only the nodes topological proximity, but also their relative position.
- the acceptation step verifying epipolar geometry of the most probable loop-closure has been replaced by a validation stage which select the loop-closure that present the smallest 2D motion in images among all the hypothesis whose probability is above a threshold.

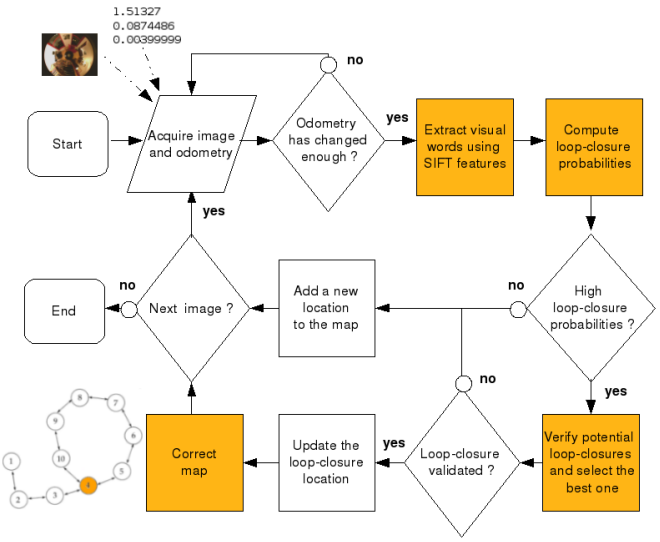

Fig. 3. Processing diagram of the topo-metric map building.

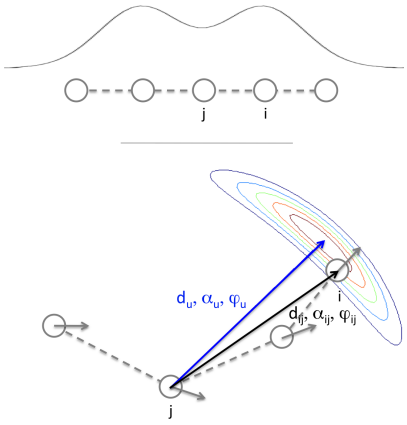

Fig. 4. Top: the original evolution model, assuming only temporal consistency of positions. Bottom: the new evolution model including odometry and relative node positions (see text for details).

- the relative position between nodes is saved on each link of the graph and a fast relaxation algorithm is applied each time a loop-closure is detected.

Beside the capacity to guide the robot because the map is more accurate and incorporate consistent geometric information, the addition of this second sensor makes also the system more robust notably in case of vision system failure (i.e. device problem, sensor occlusion, strong lighting change, dark areas).

C. A new map and a relaxation algorithm

The topological map is constituted of a set of nodes associated with an image and linked by edges. We have integrated metrical information in two forms in order to produce a topo-metric map. First, each node is associated with an absolute pose in the map (x, y, θ), where x and y are the 2D position coordinates and θ an angle representing the direction of the robot when the image was taken. Secondly, the edges are associated with a relative position between two nodes defined by (d, α, ϕ), where d and α are the polar coordinates of the second node in the coordinate space of the first, and ϕ is the difference angle between the two nodes direction.

As shown on Fig. 2 and 7 (Top) the geometric consistency of the map deteriorates over time. When a loop-closure is detected the robot is assumed to have returned at the position of a previous passing. By constraining two nodes to have the same position (Fig. 2, Bottom) we can correct the odometry drift. To do so, we choose to apply a relaxation algorithm called Tree-based network optimizer (TORO) [15], because of its speed and its high efficiency. It is called when loop-closure is found to estimate the consistent node configuration which maximally satisfy the odometry constraints between nodes. This algorithm is really fast to optimize the kind of map we use which are very simple in regards to constraints. As an example it takes less than 3 seconds for 10000 nodes, 800 constraints and 100 iterations.

D. An evolution model using odometry information

In the original framework, the evolution model used to obtain the prediction given the *a priori* probability applied a diffusion of the probability over the neighboring locations in the graph. The weight was defined as a sum of Gaussian centered on the current location (Fig. 4, Top). This diffusion was done in all directions without preference, because it only assumes that the neighboring images in time are close together, without any information about the real robot motions. Assuming such a model, the loop-closure probability tends to spread out, the system loose some responsiveness and results are not temporally consistent. Because a reliable metrical information is now available, we integrated odometry in the evolution model to predict more precisely the evolution of the probability and therefore to enhance the reactivity and above all the accuracy of loop-closure detection. Thus, starting from a given node, we distribute the probability to each neighboring location in the map depending on the deviation of these nodes relative positions with the robot displacement since the last update d_u, α_u, ϕ_u measured by odometry (Fig. 2, Bottom).

We used the standard motion model for robot odometry [29], assuming Gaussian noise on the robot displacement measured in polar coordinates:

$$p(d, \alpha, \phi | d_u, \alpha_u, \phi_u) = G_{\mu_d, \sigma_d}(d - d_u) G_{\mu_\theta, \sigma_\theta}(\alpha - \alpha_u) G_{\mu_\phi, \sigma_\phi}(\phi - \phi_u)$$

where d, α gives the odometry displacement in polar coordinates in the frame of the previous robot position and ϕ is the variation of robot direction during movement. $G_{\mu,\sigma}(X)$ is the Gaussian distribution of mean μ and variance σ^2.

The evolution model becomes: $p(S_i | S_j, u_t, M) =$

$$G_{\mu_d, \sigma_d}(d_{ij} - d_u) G_{\mu_\theta, \sigma_\theta}(\theta_{ij} - \theta_u) G_{\mu_\phi, \sigma_\phi}(\phi_{ij} - \phi_u)$$

where $u_t = d_u, \theta_u, \phi_u$ gives the odometry displacement and $d_{ij}, \theta_{ij}, \phi_{ij}$ is the relative position between nodes i

and j. The substitution makes the prediction of the *a posteriori* probability more accurate, improving robustness and responsiveness of the algorithm. The original algorithm required several corresponding frames before detection, and sometimes gives some bad results (consecutive loop-closure on the same node). By including the use of odometry two less images are required to detect, and the temporal consistency of the detection is well improved.

E. Loop-closure validation

In our previous work we verified the epipolar constraint by computing the essential matrix for the loop-closure candidate showing the higher probability. If the loop-closure was not validated, verification follows in the descending order of loop-closure probability. This strategy gives confidence in probability results and is not robust enough for our use: often the loop-closure which shows the highest probability is visually correct but is not the one corresponding to the smallest robot displacement. To find the previous position the closest to the current one we choose to use the results of the Bayesian filter as a first step to select potential loop-closure locations. On a next step, we verify all the hypothesis above a threshold with a 2D motion computation based on the SIFT keypoints and we select the loop-closure which shows the smallest translation. In order to discard outliers, the 2D motion (translation and rotation in image plane) is computed using RANSAC, accepting the result only if the number of matching points is above a threshold (30).

This new validation stage makes the method calibration free, and can be used with perspective or omnidirectional camera. According to the model of camera, we can distinguish two different cases for the graph constraints:

- Perspective camera: we assume the robot has returned to the same position in the same direction so we constrain the two positions to be the same.
- Omnidirectional camera: the robot has returned to the same position but with a different direction. In the relaxation algorithm we constrain the nodes to have the same position with a difference between the two directions that is equal to the angle between the two omnidirectional views.

This validation is more accurate but also more restrictive compared to the previous one because it does not accept images if the motion between the two is too important. This method therefore requires that the robot come back close to a previously visited locations, which is well suited for indoor environments where doors for example are a required route to go from one room to an other. Outdoor environment have also been tested successfully but usually provide less loop-closure detection as the path are often less constrained.

V. Results and Discussion

To demonstrate the quality of the approach we have used data acquired with a Pioneer 3 DX mobile robot and some data set used by the SLAM community (omnidirectional

Fig. 5. Examples of loop-closure accepted at a door crossing in Gostai sequence. On the left, two matching images accepted by the epipolar validation module (more than 1 meter between position), on the middle and on the right, images accepted using the new validation stage. The new method is more restrictive but ensure accuracy of detection and therefore a quality of the resulting topo-metric map.

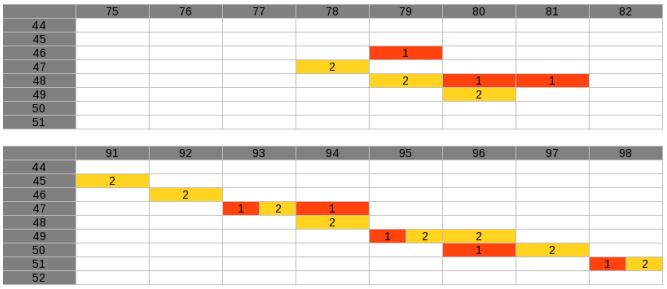

Fig. 6. Comparing loop-closure detection results with the two method (Gostai sequence). In red (1) using the previous approach with diffusion transition model and epipolar check without threshold. In Yellow (2) using the proposed approach with odometry transition model and the new validation check (see text for details).

home sequence[1], and Oxford city center sequence[2]). Concerning our data set, the robot was guided to do some loops in indoor environments showing strong perceptual aliasing conditions. The images and the odometry information were taken each time the robot moves at least 25 cm or turn of at least 10 degrees. This sampling rate is largely enough to describe the environment without saving too much redundant information, it corresponds to an average acquisition time of one image each 0.7s and an average speed of the robot about 0.4 m/s. The computer used for experimentation was an Intel Xeon 3Gh, and the images size 320x240. The path of the experiments (for the sequences providing odometry) and the resulting corrected maps are shown on Fig. 2, 7, 8, and 9.

Table I shows the improvements obtained with the new approach (LCD New) over the previous one (LCD Old). We can see that the detection rate is improved and that the false alarms are nearly suppressed. Also, our improvements produce precise loop-closure detection that always lead to maps consistent with the real world. Even if some false alarms exist, they are produced by very close location, leading to maps with the correct global topology but with local drift in some situations such as very long corridors (Fig. 9). The last sequence "city" which was outdoor presents

[1] http://www2.science.uva.nl/sites/cogniron/fs2hsc/Data/Home2/run1 [30]
[2] Oxford Mobile Robotics Group. City center [4]

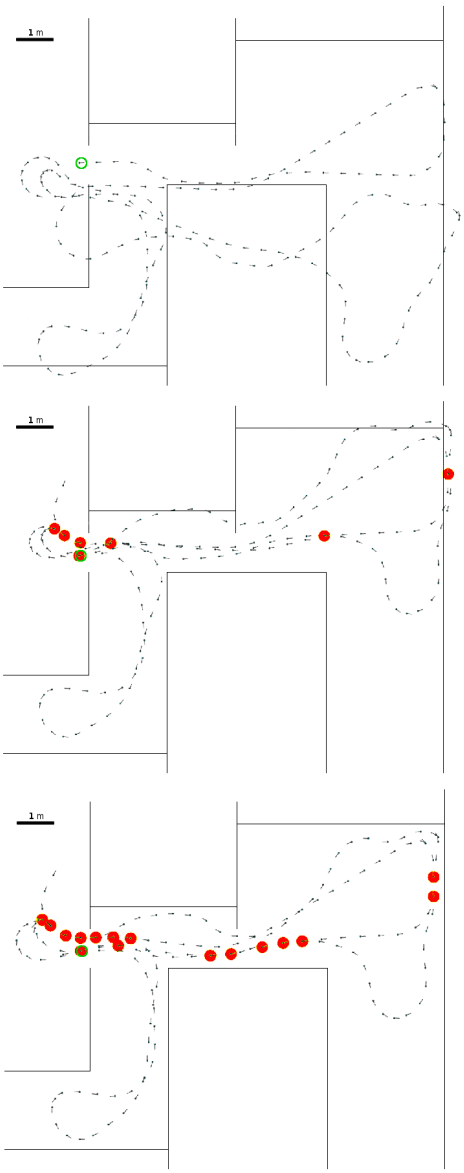

Fig. 7. Example of topo-metric mapping result (sequence Office). (a) raw odometry, (b) map using the Gaussian sum transition model and the epipolar geometry. (c) results using the odometry transition model and the new validation step. The landmark is the starting location just in front of the first door.

TABLE I

SUMMARY OF LOOP-CLOSURE DETECTION ON DIFFERENT SEQUENCES.

	Museum	Gostai	Lab	Home	City
Images	112	169	350	1400	1237
Odometry	Yes	Yes	Yes	Yes	**No**
Omni	No	No	No	**Yes**	No
CPU Time	42s	70s	208s	22min	18min
CPU Time/image	0.37s	0.41s	0.59s	0.94s	0.87s
Figure	1-2	7	9	-	-
LCD Truth	14	25	9	≈300	≈500
LCD Old [1]	2	7	2	229	-
Missed	64 %	52 %	55 %	23 %	-
False alarm	0 %	28 %	22 %	-	-
LCD New	14	18	7	256	84
Missed	0 %	12 %	20 %	14 %	79 %
False alarm	0 %	5 %	0 %	-	-

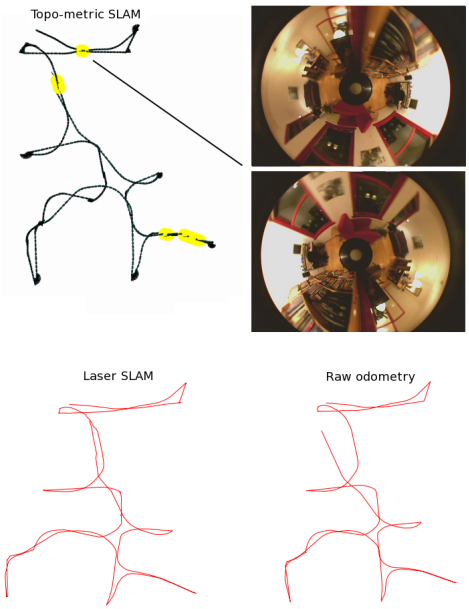

Fig. 8. Example of results of topo-metric mapping using an omnidirectional camera (sequence Home).

lower detection results mainly because the odometry was not available. Figure 6 shows some loop-closure details on the Office sequence to highlight differences between the two models. Notably, we can see that the detection rate is more important: only two consecutive similar frames are now required before effective loop-closure detection and the first validated loop-closure with the new model comes two images earlier. This figure also illustrate that the temporal consistency of the detections is recovered; there is no more gap between images and multiple loop-closure detection on the same node are discarded.

Concerning computation times, the duration of the mission "Lab" was 241 seconds and the overal processing time was 208 seconds. Extrapolating these data, we can stay on real time processing for environments up to 1400 images. It has to be noted that SURF could be used in order to replace SIFT which will divide by 2 the keypoints extraction time (SIFT 190ms, SURF 85ms for our 320x240 images). Using SURF would therefore lead to a 15% coputation time reduction as the time for SIFT extractions is about 30% of the total time.

VI. CONCLUSION AND FUTURE WORK

We have introduced in this paper a system that is able to build an hybrid topo-metric map in real-time without any camera calibration or learning stage. The developed framework combines vision-based loop-closure detection with metrical information given by the robot odometry. The odometry used to give a metrical position to each location is also used in the evolution model of the Bayesian filter to make the detection more accurate and responsive. The proposed solution is fast and robust with the use of our new validation stage which allows the use of any kind of camera (perspective or omnidirectional). The geometrical consistency between robot motion and visual perception is

Fig. 9. Example of a topo-metric map included in a laser map for reference (sequence Lab). (a) raw odometry, (b) trajectory corrected using our algorithm. The length of the trajectory was about 96 meters.

regain each time a loop-closure is detected with the use of a fast optimization algorithm. The generated hybrid topo-metric map is usable for robot guidance.

Our future work will be to optimize visual processing to further reduce computational cost and to apply this framework on mobile toy robots using wireless remote processing. The approach is well suited for this as robot guidance can be performed on-board using odometry, while remotely and asynchronously detecting loop-closure and optimizing maps for odometry correction. We are also planning to develop autonomous exploration algorithms taking into account the necessity to detect loop closure in order to be able to map large-scale environments.

REFERENCES

[1] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, vol. 24, no. 5, pp. 1027–1037, October 2008.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (slam): Part ii," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[3] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kröse, "Navigation using an appearance based topological map," in *IEEE International Conference on Robotics and Automation*, 2007.

[4] M. Cummins and P. Newman, "Accelerated appearance-only slam," in *IEEE Conference on Robotics and Automation*, 2008.

[5] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[6] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'07*, 2007.

[7] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Autonomous Robots*, vol. 12, no. 3, pp. 287–300, 2002.

[8] E. Eade and T. Drummond, "Monocular slam as a graph of coalesced observations," in *International Conference on Computer Vision*, 2007.

[9] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *IEEE International Conference on Robotics and Automation*, 2007.

[10] D. Filliat and J. A. Meyer, "Global localization and topological map learning for robot navigation," in *From Animals to Animats 7. The Seventh International Conference on simulation of adaptive behavior (SAB02)*, 2002.

[11] F. Fraundorfer, C. Engels, and D. Nistér, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[12] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Transactions on Robotics and Automation*, vol. 21, no. 2, pp. 196–207, 2005.

[13] S. Fu and G. Yang, "Uncalibrated monocular based simultaneous localization and mapping for indoor autonomous mobile robot navigation," in *Networking, Sensing and Control, 2009. ICNSC '09. International Conference on*, 2009, pp. 663 – 668.

[14] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. V. Gool, "Omnidirectional vision based topological navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.

[15] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[16] A. Howard, M. J. Mataric, and G. Sukhatme, "Relaxation on a mesh: a formalism for generalized localization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001, pp. 1055–1060.

[17] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Transaction on Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008.

[18] J. Kosecká, F. Li, and X. Yang, "Global localization and relative positioning based on scale-invariant keypoints," *Robotics and Autonomous Systems*, vol. 52, pp. 209–228, 2005.

[19] K. Kouzoubov and D. Austin, "Hybrid topologicavmetric approach to slam," in *In proceedings of the 2004 IEEE lnternatlonal Conference on Robotics and Automation*, April 2004.

[20] D. Lowe, "Distinctive image feature from scale-invariant keypoint," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based monte-carlo localisation with omnidirectional images," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 17–30, 2004.

[22] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, 2004.

[23] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. –, 2006.

[24] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2262–2269.

[25] P. Rybski, F. Zacharias, J. Lett, O. Masoud, M. Gini, and N. Papanikolopoulos, "Using visual features to build topological maps of indoor environments," in *IEEE International Conference on Robotics and Automation*, 2003.

[26] M. Saedan, L. Chee Wang, and M. Ang, "Appearance-based slam with map loop closing using an omnidirectional camera," in *Advanced intelligent mechatronics, 2007 ieee/asme international conference on*, 2007, pp. 1 – 6.

[27] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE transactions on robotics*, vol. 24, 2008.

[28] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," in *Robotics Science and Systems (RSS)*, Seattle, USA, June 2009.

[29] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[30] Z. Zivkovic, O. Booij, and B. Krose, "From images to rooms," *Robotic and Autonomous Systems*, vol. 55, pp. 411–418, 2007.

Controlling the autonomy of a reconnaissance robot

André Dalgalarrondo[a], Delphine Dufourd[a] and David Filliat[a]

[a]DGA / Centre Technique d'Arcueil
16 bis avenue Prieur de la Côte d'Or, 94114 Arcueil cedex, France

ABSTRACT

In this paper, we present our research on the control of a mobile robot for indoor reconnaissance missions. Based on previous work concerning our robot control architecture HARPIC, we have developped a man machine interface and software components that allow a human operator to control a robot at different levels of autonomy.

This work aims at studying how a robot could be helpful in indoor reconnaissance and surveillance missions in hostile environment. In such missions, since a soldier faces many threats and must protect himself while looking around and holding his weapon, he cannot devote his attention to the teleoperation of the robot. Moreover, robots are not yet able to conduct complex missions in a fully autonomous mode. Thus, in a pragmatic way, we have built a software that allows dynamic swapping between control modes (manual, safeguarded and behavior-based) while automatically performing map building and localization of the robot. It also includes surveillance functions like movement detection and is designed for multirobot extensions.

We first describe the design of our agent-based robot control architecture and discuss the various ways to control and interact with a robot.The main modules and functionnalities implementing those ideas in our architecture are detailed. More precisely, we show how we combine manual controls, obstacle avoidance, wall and corridor following, way point and planned travelling. Some experiments on a Pioneer robot equipped with various sensors are presented. Finally, we suggest some promising directions for the development of robots and user interfaces for hostile environment and discuss our planned future improvements.

Keywords: Ground robotics, autonomy, human computer interface, indoor navigation, reconnaissance robot.

1. INTRODUCTION

In November 2003, the French defense procurement agency (Délégation Générale pour l'Armement) launched a prospective program called "PEA Mini-RoC" dedicated to small unmanned ground systems. This program focuses on platform development, teleoperation and mission modules. Part of this program aims at developing autonomous functions for robot reconnaissance in urban terrain. In this context, the Centre Technique d'Arcueil (CTA) of the Délégation Générale pour l'Armement (DGA) is currently conducting studies to demonstrate the potentialities of advanced control strategies and to get a first feedback from operational forces about the use of small robotic platforms during military operations in urban terrain. Our goal is not to elaborate operational systems but to investigate several solutions[1] on experimental platforms in order to suggest requirements and to be able to build specifications for future systems. Our research focuses on robot localization,[2] autonomous map building[3,4] and robot control architecture.[5]

1.1. Overview of existing paradigms

Many paradigms have been developed for human robot interaction. They allow various levels of interaction with different levels of autonomy and dependance. In this paragraph we attempt to list briefly the main paradigms from the less autonomous (teleoperation) to the most advanced (mixed-initiative) where humans and robots can cooperate to determine their goals and strategies.

Further author information:
A.D.: E-mail: andre.dalgalarrondo@etca.fr, phone: +33 (0)1 42 31 96 59
D.D.: E-mail: delphine.dufourd@etca.fr, phone: +33 (0)1 42 31 97 07
D.F.: E-mail: david.filliat@etca.fr, phone: +33 (0)1 42 31 97 19

Teleoperation is the most basic and mature mode. In this mode the operator has full control of the robot and must shoulder total responsibility for mission safety and success. This mode is suitable in complicated or unexpected situations that no algorithm can deal with. In return this mode often means a heavy workload for the teleoperator who needs to focus his attention on the task. With *Supervisory control*[6] the operator (called the supervisor) orders a subordonate (the robot) to achieve a predefined plan. In this paradigm, the robot is merely a tool executing tasks under operator monitoring. This interaction mode can adapt to low level bandwith and high level control but needs constant vigilance from the operator who must react in case of failures (to perform manual mission re-planning for example). This mode is only suitable for static environments where planning is really effective. *Behavior-based teleoperation*[7] replaces fine-grained control by robot behaviors which are locally autonomous. In comparison to supervisory control, this paradigm brings safety and provides the robot with the opportunity to react to its own environment perception (.e.g. to avoid obstacles). Thus, it allows the operator to be more negligent. *Adjustable autonomy*[8] refers to the adjustment of the level of autononomy which has been initiated by the operator, by another system or by the system himself and while the system operates. The goal of this paradigm is generally to increase the negligence time allowed to the operator while maintaining the robot to an acceptable safety and effectiveness level. In *Traded control*[9] the operator controls the robot during one part of the task and the robot behaves autonomously during the other part of this task. This paradigm may lead to an equivalent of the kidnapped robot problem. While the robot is controlled by the operator, it can loose its situation knowledge and face difficulties when coming back to autonomous control. In *Shared control*,[10] part of the robot functions are autonomous and the remaining are controlled by the operator. It is important to notice that this mode requires constant attention from the operator. If the robot is only in charge of some safety mechanisms, this paradigm is equivalent to the *safeguarded teleoperation*. *Mixed-initiative*[11] is characterized by a continuous dialogue between the operator and the robot where both of them share decision and responsibility. *Collaborative control*[12] may be described as a restrictive implementation of mixed-initiative. In this approach, the human robot dialogue is mainly reduced to a predefined set of questions and answers.

All these paradigms may be considered as subsets of the human robot teaming domain with important overlapping. Therefore, many robotic experimentations use a mix of these paradigms.

1.2. Application to military reconnaissance robot

Military unmanned ground robots for scout or search and rescue missions have to deal with ill-known and hostile environments. Thus, in urban warfare, they face many uncertainties and intermix with friends, foes and bystanders. Technology is not mature enough to produce autonomous robots that can handle these situations on their own. Therefore robots should be used as force multipliers and risk reducer in land operations. It seems that today's best solutions have to rely on a good collaboration between humans and robots e.g. when robots act as autonomously as possible but remain under human supervision.

Here is an example of a scenario involving collaboration between soldier teams and robots. Robots could be used for reconnaissance in a building with many floors. The reconnaissance of each floor is done by a team of soldiers and robots. The robots may be carried on each floor by one or two soldiers. Another team is outside the building, surveys the environment and supervises the whole mission. At each floor, one or more robot are ahead from the soldiers and explore the floor. An operator who is responsible for robot supervision is protected by his team members so that he does not have to watch too much around him and to hold his weapon in case of imminent danger. When a team progresses into the building, the robot control is exchanged between soldiers so that the previous operator can move and follow the maneuvers safely. The coordination within the team is made by direct voice, radio or by signs. The images and data from the robots can be viewed by each team member in order to share information. Each robot can perform moving objects detection, heat and sound detection, map building and localization, navigation behaviors and give access to its data. The team which remains outside the building can also have access to this information so as to construct a global picture of the mission and to watch its evolution. In particular, it can monitor the map built at each floor by the robots and regulate the progression of the teams. When a resistance is encountered in a floor, the outdoor team can plan an action from the outside and through the appropriate windows that they have located on the same map.

Basic capacities that have been shown on many robots include obstacle detection and avoidance, waypoint and goal oriented navigation, detection of people, detection of threats, information retrieval (image, sound)

localization and map building, exploration, coordination with other robots. Most of them can be implemented on a real robot with a good reliability in reasonably difficult environnement but it is not enough to fullfill military requirements. As a partner or as an extent of a warfighter, a robot must nether increase the risk for the team – which stands for "surprise effect" cancelling and trap triggering for instance – nor impose an important workload to the human supervisor.

We think that in short term robots will not be able to handle some uncertain situations and that the most challenging task is to build a software organization which provides a pertinent and adaptative balance between robot autonomy and human control. In such teams, humans are superior in perception (especially in environment understanding), in knowledge management and in decision making. Robots can be better than humans in quantitative low-level perception, in precise localization and in displacement in confined and cluttered space. Above all, robots can stand dull, dirty and dangerous jobs. So, for a good teaming, it is desirable that robots and humans share their capacities along a two-way dialogue. This is the approximate definition of the mixed-initiative control mode but, given the maturity of today's robots and the potential danger for soldiers, we do not think that mixed-initiative could be the solution for now. It seems that adjustable autonomy, with a wide range of control modes – from basic teleoperation to even limited mixed-initiative – is the most pragmatic and efficient way.

However, independantly from the control mode, the design of an easy to handle and ergonomic operator control unit remains a major issue. A review of the research in human robot interaction is beyond the scope of this paper and of our work but we can notice that most recent work have adopted a graphical interface running on a PDA. In our development, we follow this trend and try to build a PDA-sized interface allowing a seamless and natural transition between various levels of autonomy for an indoor reconnaissance robot. This work is presented in the next paragraph.

2. PRESENTATION OF OUR WORK

Our work is based on a multiagent robot control architecture named HARPIC which was developed in[13] and which is briefly described in the next paragraph. Then we present the specific agents and the organization which allows the various control modes.

2.1. General description of HARPIC

HARPIC is an hybrid architecture (cf. figure 1) which consists in four blocks organized around a fifth: *perception processes*, an *attention manager*, a *behavior selector* and *action processes*. The core of the architecture (the fifth block) relies on *representations*.

Sensors yield data to perception processes which create representations of the environment. Representations are instances of specialized perception models. For instance, for a visual wall-following behavior, the representation can be restricted to the coordinates of the edge detected in the image, that stands for the wall to follow. To every representation are attached the references to the process which created it: date of creation and various data related to the sensor (position, focus...). The representations are stored in a table with fixed constant length, so that a round-robin mechanism keeps a given memory depth. Thus, representations are snapshots of specific landmarks in the robot's environment, the spatial and the temporal localization of which are known.

The perception processes are activated or inhibited by the attention manager and receive information on the current behavior.[14] This information is used to foresee and check the consistency of the representation. The attention manager has three main functions: it updates representations (on a periodical or on an exceptional basis), it supervises the environment (detection of new events) and the algorithms (prediction/feedback control), and it guarantees an efficient use of the computing resources. The action selection module chooses the robot's behavior depending on the predefined goal(s), the current action, the representations and their estimated reliability. Finally, the behaviors control the robot's actuators in closed loop with the associated perception processes.

The key ideas of this architecture are:

• *The use of sensorimotor behaviors linking perceptions and low-level actions both internally and externally*: the internal coupling allows to compare a prediction of the next perception (estimated from the previous perception

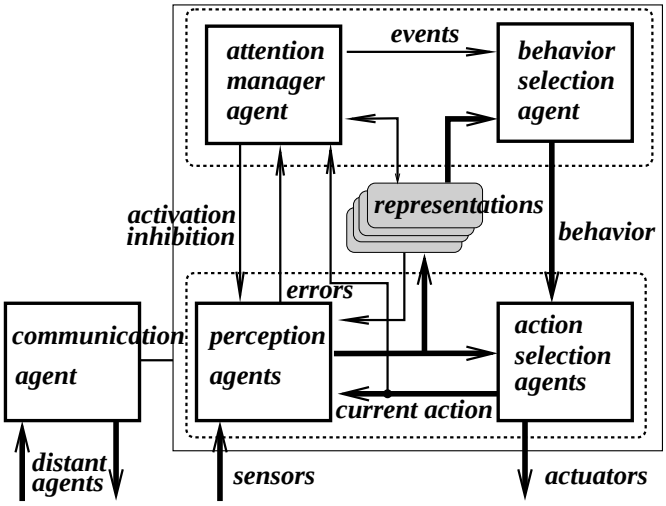

Figure 1. Functional diagram of the HARPIC architecture.

and the current control) with the perception obtained after application of the control, in order to decide whether the current behavior runs normally or should be changed; the external coupling is the classic control loop between perception and action.

• *Use of perception processes* with the aim of creating local situated representations of the environment. No global model of the environment is used; however more global and higher level representations can be built from the instantaneous local representations.

• *Quantitative assessment of every representation*: every algorithm is associated with evaluation metrics which assign to every constructed representation a numerical value expressing the confidence which can be given to it. We regard this assessment as important feature[15] since any processing algorithm has a limited domain of validity and its internal parameters are best suited to some situations only. There is no perfect algorithm that always yields "good" results.

• *Use of an attention manager*: it supervises the execution of the perception processing algorithms independently from the current actions. It takes into account the processing time needed for each perception process, as well as the cost in terms of computational resources. It also looks for new events due to the dynamics of the environment, which may signify a new danger or opportunities leading to a change of behavior. It may also trigger processes in order to check whether the sensors operate nominally and it can receive error signals coming from current perception processes. In practice, for instance with a vision sensor, the attention will focus on the illumination conditions, on the consistency between the movement of the robot and the temporal consistency of the representations, and on error signals sent by perception processes. With this information it becomes possible to invalidate representations due to malfunctioning sensors or misused processes.

• *The behavior selection module* chooses the sensorimotor behaviors to be activated or inhibited depending on the predefined goal, the available representations and the events issued from the attention manager. This module is the highest level of the architecture. It should be noted that the quantitative assessment of the representations plays a key role in the decision process of the behavior selection.

On the one hand, a representation might be more or less adapted to the current situation, depending for instance on the sensor used or on the conditions of the perception acquisition. For instance, a day camera used during night operating conditions will yield representations to which a lower confidence should be assigned a priori. The same also holds for instance for a detector relying on translational invariance while the robot has such a motion that this assumption is incorrect.

On the other hand, some representations might be more interesting for some behaviors or might provide an improved help to choose between several behaviors (e.g. a wall-following behavior needs information on contours rather than on velocity vectors, while a tracking behavior has the opposite needs). Therefore every behavior also weighs each representation depending on its direct usability, and this weight is combined with the intrinsic assessment of the representation.

• *The action selection module* regroups the lower-level controllers operating on the actuators. It uses valid representations in order to compute the control laws.

2.2. HARPIC implementation

Fundamental capacities of our architecture encompass modularity, encapsulation, scalability and parallel execution. To fulfill these requirements, we decided to use a multi-agent formalism that fits naturally our need for encapsulation in independent, asynchronous and heterogeneous modules. The communication between agents is realized by messages. Object oriented language are therefore absolutely suited for programming agents: we chose C++. We use POSIX threads to obtain parallelism: each agent is represented by a thread in the overall process. For us, multi-agent techniques is an interesting formalism and though our architecture could be implemented without them it led to a very convenient and scalable framework.

All the agents have a common structure inherited from a basic agent and are then specialized. The basic agent can communicate by sending messages, has a mailbox where it can received messages and runs its own process independantly of orther agents. Initially, all present agents have to register to a special agent called the administrator that record all information about agents (name, type, representation storage address...). All these data can be consulted by any agent. Then, when an agent is looking for another one for a specific job to do, it can access to it and to its results. It is for example what is happening when an action agent has to use a representation coming from a perception agent.

Perception and action agents follow the global scheme. The action agent is activated by a specific request coming from the behavior selection agent. The selection orders him to work with a perception agent by sending its reference. The action agent sends in turn a request to the proper perception agent. Perception agents are passive, they only run upon request, perform a one shot execution and then wait for a new message. Furthermore, a perception agent can activate another agent and build a more specific representation using its complementary data. Many action and perception agents run at the same time but most are waiting for messages. Only one behavior (composed of a perception agent and an action agent) is active at the same time. Within a behavior, it is up to the action agent to analyze representations coming from the perception agent and to establish the correct control orders for the platform.

The attention agent supervises the perception agents. It has a look-up table where it can find the types of perception agents it has to activate depending on the current behavior. It is also in charge of checking the perception results and of declaring new events to the behavior selection agent when necessary. The advantage of this organization is detailed in previous papers.[5, 16]

The selection agent has to select and activate the behavior suited for the robot mission. This agent may be totally autonomous or constitute the process that runs the human computer interface. In this work, it is the second case and this agent is detailled in paragraph 2.4.

We use two specific agents to link our architecture to hardware. The first one is an interface between the software architecture and the real robot. This agent is awaiting a message from an action agent before translating the instructions into comprehensible orders for the robot. Changing the real robot require the use of a specific agent but no change in the overall architecture. The second agent acquires images from the grabber card at a regular rate and stores them in computer memory which can be consulted by perception agents.

Finally, we use a specific agent for IP communication with distant agents or other architectures. This agent has two running loops: an inner loop in which it can intercept messages from other agents belonging to the same architecture, and an external loop to get data or requests from distant architectures. This agent surpervises the (dis)appearance of distant agents or architectures. It allows the splitting of an architecture on many computers or the communication between several architectures. For example, this agent is useful when agents are distributed between the robot onboard computer and the operator control unit.

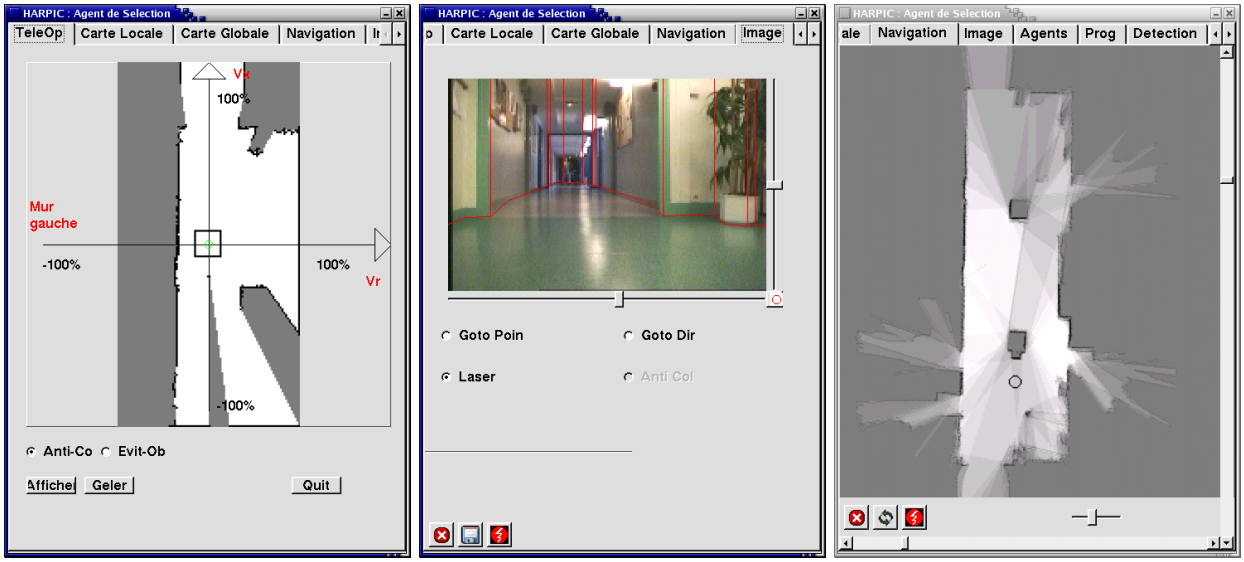

Figure 2. Interface for laser-based (left), image-based (center) teleoperation and goal navigation (right).

2.3. Agents for SLAM and path planning

Map building is performed by a perception agent that takes laser sensor data and odometry data as input and outputs a representation which contains a map of the environment made of registered laser scans. The map building technique is based on the simple and efficient mapping algorithm[17] that makes use of histogram correlation for scan matching. This algorithm can handle fairly large environments and is quite robust to large cycle closure. It proves to be very robust as long as some straight walls are present in the environment but may fail in extremely cluttered area. This agent is executed whenever new laser data are available (e.g. 5 Hz), but it adds data to the map only when the robot has moved at least one meter since the last map update.

Localization is performed by a perception agent that takes odometry, laser data and the map representation as input and outputs a representation containing the current position of the robot. In its current implementation, this agent takes the position periodically estimated by the mapping algorithm and interpolates between these positions using odometry data to provide anytime position of the robot. This agent is executed upon request by any other agent that has to use the robot position (e.g. mapping, planning, human computer interface...). Despite the fact that it is not useful in the current implementation, we have chosen to separate the localization agent from the map-building agent so as to be easily able to use either pre-recorded maps or other localization methods for comparison and evaluation purposes. For example, we are currently implementing a particle filter approach to localization[18] in order to evaluate the potential gain in robustness brought by this method.

Finally, path planning is carried out by an action agent that takes the map and position representations as inputs and outputs motor commands that drive the robot toward the current goal. This agent first converts the map into an occupancy grid and using a value iteration algorithm it computes a potential that gives for every cell of the grid the length of the shortest path from this cell to the goal. The robot movements are then derived using gradient descent on this potential from the current robot position. The path planning agent is used in the *Navigation* control mode of the operator control unit (described below).

2.4. HCI agent

In this implementation, our human computer interface is a graphical interface managed by the behavior selection agent of HARPIC. It is designed to use a small touch-screen of 320x240 pixels such as the one that equipped most personal digital assistant (PDA). With this interface, the user has to select a screen coresponding to one of

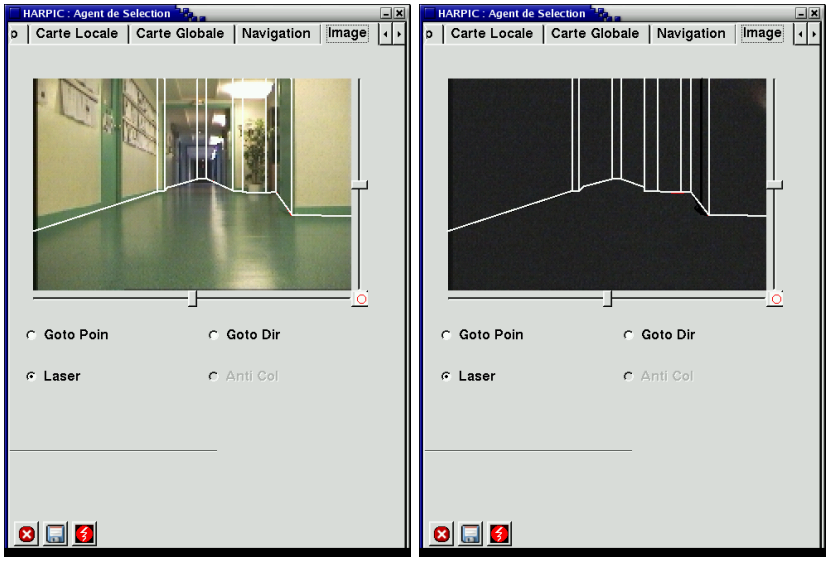

Figure 3. Image screen in normal (left) and in low-light condition (right) with overlaid polygonal view.

the control mode he wants to activate or a screen showing the environement measures and representations built by the robot. These screens are described below.

The *Teleop* screen corresponds to a teleoperation mode where the operator controls the translational and the rotational speed of the robot (see figure 2 (left)) by defining on the screen the end of the speed vector. The free space measured by the laserscanner can be superimposed on the screen and it appears in white color. The operator can also activate anti-collision and obstacle avoidance functions. Messages announcing obstacles are also displayed. This screen allows full teleoperation of the robot displacement (with or without seeing it thanks to the laser free space representation) as well as safegarded teleoperation. As a result the operator has full control on the robot in order to trade with precise movement in cluttered space or with autonomous mobility failure. In countrepart, he must keep defining the robot speed vector otherwise it stops and waits.

The *Image* screen allows to control the robot by pointing at a location or defining a direction within the image acquired by the onboard robot camera. As the previous mode, this one enables full control or safeguarded teleoperation for the robot displacement. Two sliders operate the pan and tilt unit of the camera. When the *GoTo XY* function is enabled, the location selected by the operator in the image is translated into a robot displacement vector by projection on the ground plane with respect to the camera calibration and orientation angles. The *Direction* function moves the robot when the operator defines the end of the speed vector on the screen. The selectable *Laser* function draws a polygonal view of the free space in front of the robot (in an augmented reality way) which is built from the projection of the laser range data in the image. This *tron-like* representation allows to control the robot in the image whenever there is no sufficient light for the camera. Incidentally, this function provides a visual checking of the good correspondance between the laser data and the camera data. Figure 3 illustrates the effect of this function. If the *GoTo XY* or *Direction* functions are not enabled and if the robot is in any autonomous mode, this screen can be used by a operator to supervise the robot action by viewing the images of the onboard camera. However, it can stop the robot in case of emergency.

The *Navigation* screen shows a map of the area already explored by the robot. In this control mode, the operator has to point out a goal location in the map to trigger an autonomous displacement of the robot until that goal. The location can be changed dynamically (whenever the previous goal has not been reached). The planification process is immediate (about some seconds). When new areas are discovered, the map is automatically updated. As shown in figure 2 (right), the map is an occupancy grid where bright areas correspond to location which have been observed as free more often than darkest area. Three sliders can translate or zoom

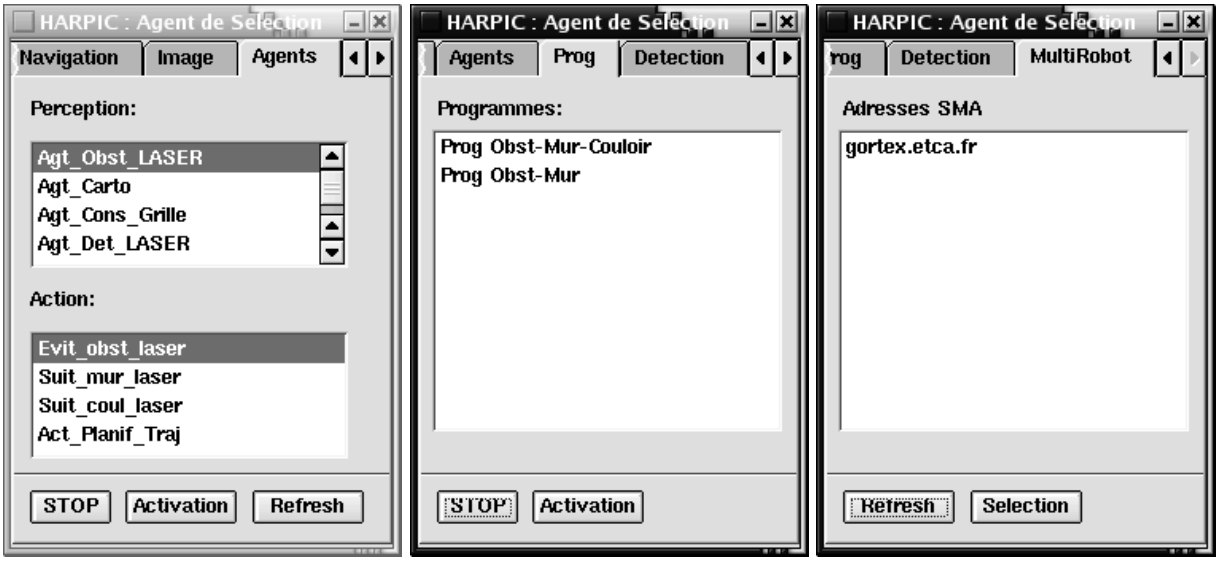

Figure 4. Interface for agent (left), program (center) and robot (right) selection.

the displayed map. This is an autonomous control mode where the operator can select a goal and forget the robot.

The *Agents* screen allows the composition and the activation of behaviors by selecting an appropriate pair of perception and action agents (see fig. 4 (left)). For example, it allows to execute behaviors like obstacle avoidance, wall following or corridor following with different perception or action agents (each one corresponding to a specific sensor or algorithm) for a same behavior. For example, a wall following behavior can result from a perception agent using the camera, from another one using the laserscanner and from various algorithms. This control mode correspond to the behavior-based teleoperation paradigm. However this screen has been mainly designed for expert users and development purpose. It lacks simplicity but it will be easily reduced to a small number of buttons when the most effective behaviors set will be determined.

The *Prog* screen corresponds to predefined sequences of behaviors. For example, it enables the robot to alternate obstacle avoidance and wall or corridor following when respectively an obstacle, a wall or a corridor appears in the robot trajectory (see fig. 4 (center)). This example is a kind of sophisticated wander mode. More generally, this mode allows automous tasks that could be described as a sequence of behavior like exploration or surveillance where observation and navigation behaviors are combined. The list of these sequences can be easily augmented to adapt the robot to a particular mission. In this control mode, the robot is autonomous and the operator workload could be null.

A *MultiRobot* screen allows the operator to select the robot which will be controlled by his control unit. Indeed, our interface and sofware architecture is designed to adress more than one robot. In a platoon with many robots this capacity may give way to the sharing of each robot data or representation and to the exchange of robot control between soldiers.

The *Local Map* and *Global Map* screens show the results of the SLAM agents described in 2.3 (see fig. 5 (left and center)). The first one is a view of the free zone determined on each laser scan. The second one displays the global map of the area explored by the robot and its trajectory. The circles on the trajectory figure the location where the SLAM algorithm has added new data. The current position of the robot is also shown on the map. As on some others screens, sliders allow the operator to translate and to zoom the display. These screens may be used when the robot is in any autonomous mode to supervise its movements.

The *Detection* screen displays moving objects trajectories in the map built by the robot (see fig. 5 (right)). This screen stands for surveillance purpose. The algorithm used is based on movement detection on laser range

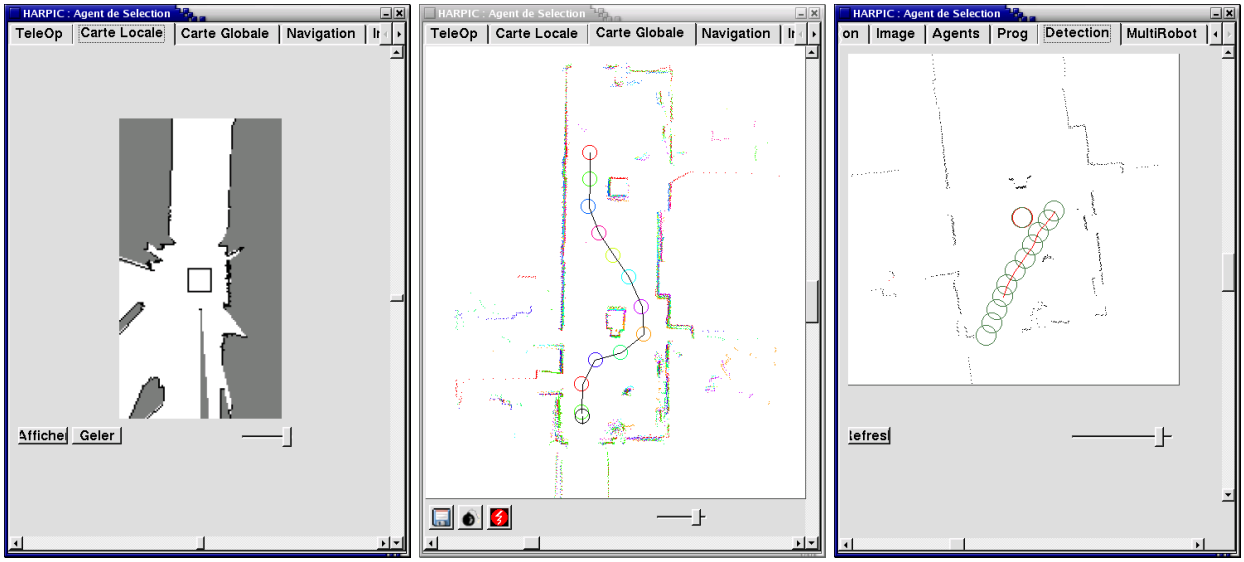

Figure 5. Interface for local map (left), global map (center) and moving object detection (right).

data and Kalman filtering. As it refers to the map built and updated by the robot, the detection and tracking of moving objects can be done while the robot is moving. This screen shows that this interface is not limited to displacement control of the robot but can be extended to many surveillance tasks.

The transition between any autonomous or teleoperation screens causes the ending of the current action or behavior. These transitions have been designed to appear natural to the operator. However, when one of these modes is actived, it is still possible to use the screens that display robot sensors data or representations without disactivating them. This feature is valid for the operator control unit but also for other control units. Thus, in a soldier team images and representations from a given robot can be viewed by a team member that is not responsible for the robot operation.

2.5. Experimentation

The robot we used both in indoor and outdoor experiments is a Pioneer 2AT from ActivMedia equipped with sonar range sensors, color camera (with motorized lens and pan-tilt unit) and an IBEO laser range sensor. On board processing is done by a rugged Getac laptop running Linux and equipped with IEEE802.11 wireless link, frame grabber card and Arcnet card for connection to the laser (cf. figure 6). We also use another laptop with the same wireless link that plays the role of the operator control unit. The agents of the robot control architecture are distributed on the two laptops. We did not use any specialized hardware or real-time operating system.

Several experiments have been conducted in the rooms and corridors of our building and have yielded good results. In such an environment with long linear walls or corridors, the automous displacement of the robot using the implemented behaviors is effective. However, in this particular space, a few limitations for the SLAM process have been identified. They mainly come from the laser measurement when the ground plane hypothesis is not valid and in the presence of glass surfaces.

The largest space we have experimented so far was the exhibition hall of the JNRR'03 conference.[19] Figure 7 shows the global map and the robot trajectory during this demonstration. It took place in the machine-tool hall of the Institut Français de Mécanique Avancée (IFMA) and in the presence of many visitors. As could be seen on figure 7, these moving objects introduced some sparse and isolated edges on the map but did not disturb the map building process. The robot travelled an area about 60×60 m large with loops and abrupt heading changes. The robot displacement was mainly done in the safeguarded teleoperation mode because the building lacked main structures and direction for the robot to follow and because of the presence of people.

Figure 6. The experimental platform.

These experiments have revealed some missing functions in our interface (e.g. "return to starting point" behavior, manual map management such as the limitation of the map space, etc) but no deep change requirement in the software architecture have been discovered.

2.6. Future work

Our human computer interface runs on a PC laptop with Linux and Qt C++ toolkit for the graphic interface. It is currently being integrated on a PDA with Linux and Qtopia environment. New functions and behaviors will appear within the next months. We are working on behaviors for people following that are based on image and laser data processing. They will allow an operator to make a robot follow him or another team member. The integration of a catadioptric camera to cope with the narrow field of view of the actual onboard camera is also on the run. Next we plan to develop autonomous behaviors for exploration, go-back-home and for assistance in narrow space crossing like doors. In the meantime, we keep improving existing behaviors so as to make them as robust as possible. Development of more multirobot capacity, interaction and cooperation are also planned as a second step. Our work focuses on function development and integration and not on ergonomy and usability of the interface. We think that such studies are necessary but will have to be conducted after the definition of the functions of the interface and their transition mechanisms.

3. SUMMARY

This article describes our work concerning the development of an effective software to control a reconnaissance robot. We have created a robot control architecture based on a multiagent paradigm that allows various levels of autonomy and interaction between an operator and the robot. According to the state of the art in robot autonomy in hostile environment we think that *adjustable autonomy* is the most pertinent choice for the next generation of military robots. Our work shows that many control modes with seamless transitions can be fairly integrated in a quite simple operator control unit. It has also confirmed the good behavior of our software architecture HARPIC and the great advantage of the flexible multiagent approach when adding many functions. Moreover this kind of interface is a good mean to experiment, evaluate or demonstrate autonomous robot behaviors and human robot interaction. We hope that it will soon enable us to gather more military requirements and to improve the specification of future systems.

Figure 7. Map and robot trajectory generated during a JNRR'03 demonstration in the IFMA hall. Each circle on the robot trajectory represents the diameter of the robot which is about 50 cm.

ACKNOWLEDGEMENTS

This work was entirely performed at the Geomatics-Imagery-Perception department of the Centre Technique d'Arcueil within the DGA. We would like to thank S. Moudere, E. Sellami, N. Sinegre, F. Souvestre, D. Lucas, R. Cornet and G. Calba for their contributions to the software development of this work.

REFERENCES

1. D. Luzeaux, A. Dalgalarrondo, and D. Dufourd, "Autonomous small robots for military applications," in *UVS Tech 2001*, (Belgian Royal Military Academy, Brussels, Belgium), 6-7 december 2001.
2. D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots: I. a review of localization strategies," *Cognitive Systems Research* **4 (4)**, pp. 243–282, december 2003.
3. D. Dufourd, "Autonomous construction of indoor maps with a mobile robot," in *SPIE* 15^{th} *Annual Symposium, AEROSENSE'01, Unmanned Ground Technology III*, (Orlando, FL), april 2001.
4. D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots: Ii. a review of map-learning and path-planning strategies," *Cognitive Systems Research* **4 (4)**, pp. 283–317, december 2003.
5. D. Luzeaux and A. Dalgalarrondo, *Hybrid architecture based on representations, perception and intelligent control*, ch. Studies in Fuziness and Soft Computing: Recent Advances in Intelligent Paradigms and Applications. ISBN-3-7908-1538-1, Physica Verlag, Heildelberg, 2003.
6. P. Backes, K. Tso, and G. Tharp, "The web interface for telescience," in *IEEE International Conference on Robotics and Automation, ICRA'97*, (Albuquerque, NM), 1997.
7. M. Stein, "Behavior-based control for time-delayed teleoperation," Tech. Rep. 378, GRASP Laboratory, University of Pennsylvania, 1994.
8. G. Dorais, R. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost, "Adjustable autonomy for human-centered autonomous systems," in 6^{th} *International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Adjustable Autonomy System*, 1999.
9. D. Kortenkamp, R. Bonasso, D. Ryan, and D. Schreckenghost, "Traded control with autonomous robot as mixed initiative interaction," in 14^{th} *National Conference on Artificial Intelligence*, (Rhode Island, USA), july 1997.
10. T. Röfer and A. Lankenau, "Ensuring safe obstacle avoidance in a shared-control system," in *Seventh International Conference on Emergent Technologies and Factory Automation, EFTA'99*, (Barcelonna, Spain), 1999.
11. G. Ferguson, J. Allen, and B. Miller, "Trains-95: Towards a mixed-initiative planning assistant," in *Third International Conference on AI Planning Systems, AIPS-96*, 1996.
12. T. Fong, C. Thorpe, and C. Baur, "Collaborative control: a robot-centered model for vehicle teleoperation," in *AAAI Spring Symposium on Agents with Ajustable Autonomy, Technical report SS-99-06*, (Memlo Park), 1999.
13. A. Dalgalarrondo, *Intégration de la fonction perception dans une architecture de contrôle de robot mobile autonome*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, janvier 2001.
14. A. Dalgalarrondo and D. Luzeaux, "Dynamic selection of perception process on an autonomous robot," in *SPIE* 13^{th} *Annual Symposium, AEROSENSE'99, Vol. 3691*, (Orlando, FL), april 1999.
15. A. Dalgalarrondo and D. Luzeaux, *Assessment and applications assessment of image processing algorithms as the keystone of autonomous robot control architectures*, ch. Imaging and Vision Systems, Advances in Computation: Theory and Practice. NOVA Science Books, New York, 2001.
16. A. Dalgalarrondo and D. Luzeaux, "Introducing attention in a behavior-based robot control architecture," in *European Control Conference, ECC'99*, (Karlsruhe, Germany), september 1999.
17. T. Röfer, "Using histogram correlation to create consistent laser scan maps," in *IEEE International Conference on Robotics Systems (IROS-2002)*, (EPFL, Lausanne, Switzerland), 2002.
18. S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence* **128 (1-2)**, pp. 99–141, may 2001.
19. "Quatrièmes Journées Nationales de Recherche en Robotique (JNRR'03)," (Clermont-Ferrand, France, http://lasmea.univ-bpclermont.fr/jnrr03/), 8-10 octobre 2003.

Multi-sensor semantic mapping and exploration of indoor environments

Islem Jebari, Stéphane Bazeille, Emmanuel Battesti, Hassene Tekaya, Marius Klein, Adriana Tapus, David Filliat
Cognitive Robotics Theme, ENSTA ParisTech
Paris, France

islem.jebari@ensta-paristech.fr

Cédric Meyer, Sio-Hoï Ieng, Ryad Benosman
Institut Des Systemes Intelligents et de Robotique (ISIR-CNRS)
Paris, France

Eddy Cizeron, Jean-Charles Mamanna, Benoit Pothier
GOSTAI S.A.S
Paris, France

Abstract—**The human perception of the external world appears as a natural, immediate and effortless task. It is achieved through a number of "low-level" sensory-motor processes that provide a high-level representation adapted to complex reasoning and decision. Compared to these representations, mobile robots usually provide only low-level obstacle maps that lack such high-level information. We present a mobile robot whose goal is to autonomously explore an unknown indoor environment and to build a semantic map containing high-level information similar to those extracted by humans and that will be rapidly and easily interpreted by users to assess the situation. This robot was developed under the Panoramic and Active Camera for Object Mapping (PACOM)[1] project whose goal is to participate in a French exploration and mapping contest called CAROTTE[2]. We will detail in particular how we integrated visual object recognition, room detection, semantic mapping, and exploration. We demonstrate the performances of our system in an indoor environment.**

Keywords: SLAM; object detection; autonomous exploration; semantic mapping;

I. INTRODUCTION

The Panoramic and Active Camera for Object Mapping (PACOM) project addresses the understanding of how an autonomous embodied system can build and extract information from sensory and sensory-motor data and generates plans and actions to explore and navigate in typical indoor environmental settings. In particular, we seek to extract high-level semantic information that is easy to understand and interesting to the robot users such as surrounding objects and the environment structure. To achieve this goal, the system requires different sensing modalities and also needs to act in order to improve its understanding of the environmental situation or to disambiguate its interpretation.

The project goal is to participate in the CAROTTE challenge that proceeds over three years with an increase in the difficulty over the years. The competition between 5 selected teams takes place in an arena of approximately 120m² where objects are laid. The environment contains several rooms typically 10 or more, with variable grounds and various difficulties (fitted carpet, tiling, grid, sand, stones,…). Several kinds of objects are present, either isolated or gathered, in multiple specimens, which must be detected, located, and identified or characterized by the robot. The objects used in the competition are: computers, boxes, paperboards, journals, books, telephones, keys, pens, staplers, cables, lamps, weapons, ammunition, cans, bottles, plants, cameras, radios, ventilators, balloons, robots. The complete description of the challenge can be found on the website[2].

After a review of the related work, this paper will detail the hardware and software architecture of our robot in Section III, describe the software components used in Section IV before giving experimental results in Section V.

II. RELATED WORK

This project involves the integration of several research areas for which we will make a short survey: localization and mapping for mobile robots, object detection and recognition and exploration of unknown environments.

The problem of Simultaneous Localization and Mapping (SLAM) of an unknown environment by a mobile robot is the subject of a lot of research for more than 20 years. Today, very robust solutions exist for SLAM in planar environments using 2D scanning laser sensors to the point where several effective commercial or open source software packages are available. Research on SLAM using laser sensors is now focused on 3D SLAM [1] which is made possible by combining multiple planar laser scanners with different orientations or by using a planar laser sensor rotating around a second axis. The most active research area in SLAM is probably visual SLAM with

[1] The PACOM project is supported by DGA in the frame of the "CAROTTE" competition and funded by ANR under the subvention 2009 CORD 102.

[2] CAROTTE is organized by the French research funding agency (ANR) and the French armament procurement agency (DGA). Website: <http://www.defi-carotte.fr>

different approaches including metric landmark-based SLAM using monocular vision [2], panoramic vision [3], or stereo-vision [4] and topological SLAM using panoramic [5] or monocular vision [6]. However, in an autonomous exploration context, these approaches are limited, as they do not usually provide an estimation of free space for robot movement when using panoramic or standard camera, and are not reliable when facing texture-less walls in indoor environment when using stereovision. Vision in general is also limited by the need of ambient light, which could be absent in some environments of interest. Finally, several researchers integrated laser sensor and panoramic vision [7], which provides the robustness of obstacle sensing by laser and very informative visual data. Beside these approaches that are mainly directed toward low level robot localization and navigation, several other approaches have been proposed to introduce higher-level semantic information into maps. This includes the classification of space into different categories such as rooms, corridors [8], roads, buildings [9] and the addition of objects in a hierarchical map representation. One related problem in computer vision is the use of contextual information to enhance object detection and recognition [10].

In the field of object detection, several approaches enable the robot to recognize objects using vision in complex and realistic environments. Many of these approaches are based on robust, scale invariant keypoint detection and recognition (e.g., SIFT [11], SURF [12]) or on machine learning to achieve fast and reliable keypoint recognition [13]. Beside individual object recognition, these features can be used for object class recognition using the "bag of keypoints" approach [14]. Object recognition is also possible using 3D information taken from a laser scanner [1].

The exploration of an unknown environment using a 2D scanning laser sensor is often performed using the frontier-based exploration algorithm [15] or an exploration strategy choosing the next best position for the robot given the utility of this position for the mapping problem [16]. In the context of this project, the exploration should also take into account the need to search for objects. This general problem of optimally covering the whole environment with several viewpoints is referred to as the "art gallery problem" [17] and is known to be NP-hard. However, several practical solutions exist, for example by optimizing the probability of finding the target given a limited number of actions [18]. It should be noted that several work on semantic mapping do not mix the problems of unknown environment exploration and object search, either by relying on a human interaction to guide exploration or by using a previously built map [19].

III. SYSTEM OVERVIEW

We developed a robot (see Fig.1 Right) based on a pioneer 3 dx from Mobile Robots Inc. The robot was fitted with 2 scanning laser range finders (one horizontal SICK LMS 200 and one vertical Hokuyo UTM 30 LX), a ring of sonar sensors, a Pan-Tilt-Zoom camera and three on-board computers. Our robot has not been optimized for the competition only as it can be used for assistive applications for which the functionalities of semantic mapping are particularly interesting.

The software architecture (see Fig.1 Left) uses the Urbi framework; a middleware for programming complex robotic systems developed and supported by Gostai[3]. Urbi is composed of a distributed component architecture (UObject), and an innovative orchestrator language (urbiScript) to coordinate all components. This language incorporates high-level features that facilitate the development of parallel and event-based applications.

For the project, we thus developed a set of UObjets in C++ carrying out the various necessary functionalities. The whole mission of the robot is implemented in urbiScript which uses these UObjets functionalities and coordinates their activation.

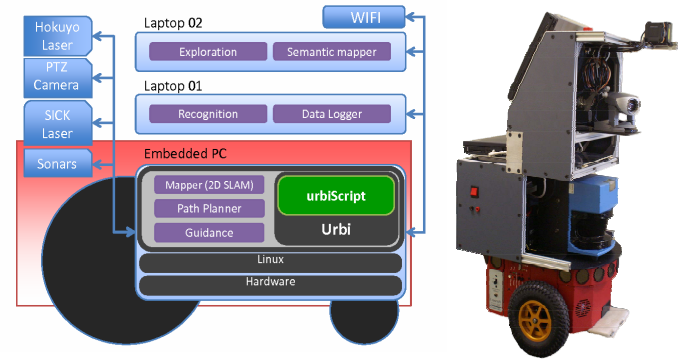

Figure 1. Left: The structure and distribution of the various processes in the robot. Right: The hardware architecture of the robot

A simplified version of the urbiScript code for the competition's mission is shown on Figure 2. It uses the UObjects described in the next section.

```
every(500ms) { logger.savePosition();}

every(300ms)
      { if(robot.isMoving())
            { image_buffer.pushBack(camera.val);}}

whenever(!image_buffer.empty())
  {recognition.localizeObjects(image_buffer.removeFront());}

while(execution time<max_time &&
!exploration.complete())
      { exploration.computeNextPosition();
        robot.goToPosition(exploration.nextPosition);
        for( var pan = pi; pan >= -pi; pan -= pi/4)
              { robot.orientCamera(pan,tilt);
             image_buffer.pushBack(camera.val);}
        mapper.updateMap();
        exploration.updateMap();

      }


robot.goToPosition([0,0,0]);

semantic.addInformationToTheMap(map);
```

Figure 2. Part of the urbiScript code implementing the robot mission. Interesting features are the use of "every" for background position and image logging during robot movement and the use of "whenever" for asynchronous image processing in order to be able to move the robot during object detection

[3] Gostai S.A.S. Website : <http://www.gostai.com>

IV. SYSTEM OVERVIEW

The main software components developed within our project are: Exploration, Object recognition and Semantic mapping. In addition to these main parts, we also integrated other components for Mapping, Path planning, Guidance and Data Logging.

Mapping performs 2D SLAM using the horizontal laser scanner and the Karto software library[4], which provides good performance and robustness in indoor environments. This library uses scan matching to correct the robot odometry drift and provides a 2D occupancy grid map. Path planning performs global path planning given the current map of the environment and carries out local obstacle avoidance of dynamic objects taking into account lasers and sonars. This module is also based on the Karto library. Guidance controls the robot to execute the computed path. Data Logging records the data and the results of the mission.

A. *Exploration*

In our context, exploration is required to completely map the environment using the horizontal laser scanner and to search for objects detected using the pan-tilt camera. We took advantage of the fact that the two sensors have similar field of view (a semi-circle in front of the robot) to integrate these two objectives into a single algorithm.

Computing an optimal set of positions to cover the environment is a NP-hard problem. We therefore use a stochastic sampling strategy inspired by the Randomized Art-Gallery Algorithm [20] to search for the next robot position that discovers the more unseen area. In order to minimize robot movements, N positions are sampled inside the free space in a semi-circled area in front of the robot (N=50 in this paper). If no reachable position or no unseen area can be found in this area, positions are sampled from the whole map to ensure global exploration.

Each sampled position is evaluated using a scoring function that takes object search and mapping into account. For object search, a score S_{obj} is computed as the size of the area visible through the camera that has not been observed yet. The visible area takes the pan-tilt capability of the camera and the minimal resolution needed for object recognition into account. The unobserved area is computed using a 2D view-map that records the area previously perceived by the robot camera. For mapping, a score S_{map} is computed as the number of frontier cells between free and unknown area [21] that are visible trough the laser sensor from the position. Finally, a score S_{dist} is computed as the inverse of the travel distance from the current position. The final score is a weighted sum of these components:

$$S = S_{obj} + \lambda_{map} S_{map} + \lambda_{dist} S_{dist} \quad (1)$$

The sampled position with the highest score is taken as the next robot position. The exploration is performed until a given percentage (98% in this paper) of the free area has been observed by the camera.

[4] Karto Robotics : Website <http://www.kartorobotics.com/>

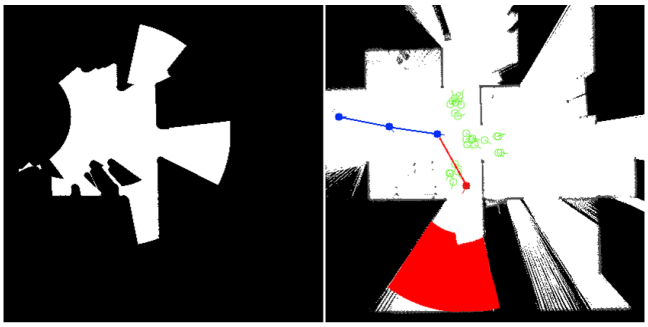

Figure 3. Illustration of the exploration strategy. The left part shows the view-map recording already seen areas. The right part shows the exploration trajectory, the sampled positions (in green) and the selected next position with the next observed area (in red)

B. *Object recognition*

The goals of the object recognition task are: first, to detect the objects in the robot's visual environment and secondly, to estimate their location in the camera reference frame. Object detection is performed using a two step method. The first step includes a fast extraction of the salient regions [22] in each image in order to segment the image to allow multiple object detection and to improve detection and localization speed by reducing the regions of interest. The second step is based on local features extraction [12, 23] and a bag of visual words approach [24]. Once an object is detected, its localization is estimated using additional information on objects or with a second step of dense features extraction [11] followed by the computation of the geometric transformation between the real object and a previously learned image. Objects to detect are known "a priori" and a database has been built containing each object over different points of view and different known distances.

a) *Image segmentation*

Segmentation is performed by computing the entropy based on the measure of color and gradient rarity [22]. Each image (Fig.4-1) is subsampled 10 times to 64x48 pixels in order to accelerate computation. The saliency map is computed (Fig.4-2) and is used to find local maximums that are passed through a region growing and fusion algorithm to find stable salient regions (Fig.4-3). The smallest regions are then filtered in order to find the candidate objects (Fig.4-4) that are represented by a rectangular bounding box.

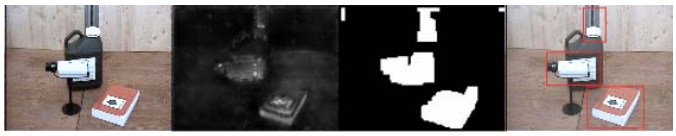

Figure 4. Steps of image segmentation : the first image is the original image taken by the robot, second is the saliency map computed on the subsampled image, third is the saliency map once thresholded and passed through region growing and fusion algorithm. The last one shows the salient regions drawn on the original image

b) Object detection

The detection algorithm is based on a bag of visual words approach [24] (Fig.5) in which images are only represented by the occurrence frequency of local features taken from a dictionary. In our approach, two dictionaries are built by extracting SURF keypoints [12] and local color histograms [23] from each image of the learning database. Descriptors are then clustered using hierarchical k-means. These tree structures [25] are used to improve the matching speed between descriptors and visual words.

Object learning is performed by computing visual words corresponding to the features extracted in the object images, and by associating to each visual word its occurrence frequency for each object. The structure obtained is a reverse index allowing fast object detection from image descriptors.

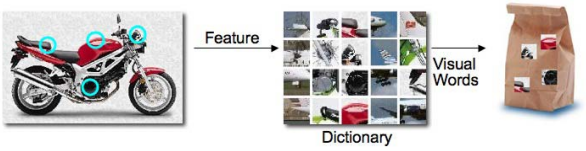

Figure 5. Bag of visual words representation

For object recognition during the robot exploration, visual words are extracted from salient regions of each image taken by the camera. The inverse index is then used with a Bayesian voting method [23] which indicates the presence of a learned object in each of the salient region.

c) Object localization

When an object has been detected in a sub-image, it has to be localized in order to be added to the semantic map. For this, we either compute the geometric transformation between the detected object and its image in the database when the object is textured, or we suppose that the object is put on the ground when the object is colored but textureless.

The geometric transformation calculates a set of matching points between the object to localize and a learned image using SIFT descriptors [11] (SURF descriptors are not dense enough to get enough correct matches on our images). Pairs of locations of matched points are given as an argument to an algorithm that computes the best homography using RANSAC method [26] in order to eliminate outliers, which increase the accuracy of the detected position of the object in the captured image. The distance of the recognized object is computed by using each inter-pair distance between the matched points to estimate scale difference between the two images. The matching method allows object localization even if they are partially hidden. Given that the density of SIFT descriptors is very low on uniformly colored objects, we assume that colored objects are put on the ground, which is quite often in the context of CAROTTE. We just have to determine the object position in the image and to intersect a line of view with the ground plane. In order to accelerate this step, we use SURF visual words already extracted by the object detection algorithm previously discussed in order to find the center of object (pink crosses in Fig.6).

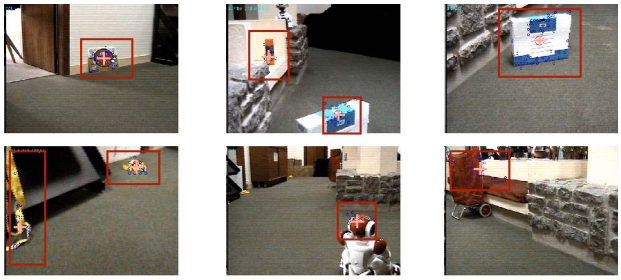

Figure 6. Result of the object localization during an exploration. Red rectangles correspond to salient regions where objects have been detected (from a set of 10 previously learned objects), blue circles are the visual words extracted in the salient regions (SURF descriptors and local color histograms) and pink crosses are the estimated position of object center used to compute object distance

As we can see in Figure 6, we have a reasonable localization of the object in the image. However, if the object is big or not totally settled on the ground, the estimated distance of the object has greater errors which could be corrected using Kalman filters on the estimated objects position as shown in the next section.

C. Semantic mapping

"Semantic mapping" adds meaningful information to the 2D map such as the 3D structure of the environment, the position, name and images of the detected objects and the rooms. The 3D structure is built as a point cloud using the second laser sensor, assuming that the "mapping" module gives a correct position. We estimate the object position by integrating multiple detections using Kalman filter [27]. The application of this method is particularly indicated in our case because we have an external estimate of the robot position. Rooms are detected in the occupancy grid through an algorithm that detects doors and analyses the resulting connected components of open space.

The 2D semantic mapping consists in calculating the coordinates of the object detected in the absolute reference frame. The object detection module gives the identifier corresponding to the object category, but it should be decided if this object has already been seen or if it is a new exemplar of the same object. For this, we calculate the jacobian matrix associated to the transformation from the image position to the map position in order to estimate the uncertainty of the position of the object in the world frame. We use the Mahalanobis distance to decide if a perception corresponds to an object already present in the map or not. Thus, for each object already placed in the map corresponding to the same category as the detected object, we compute the Mahalanobis distance with the detected object and if the minimal value of the Mahalanobis distance is higher than a certain threshold, we add the object detected as a new object; If not, we update the position of the object recognized with the Kalman filter.

For the 3D semantic mapping, we chose an approach rather similar to the object mapping. The positions of the obstacles detected by the second laser, directed vertically (with a slope of 60° compared to the horizontal one) are calculated in the world reference and are memorized. This position is calculated by supposing exact the position of the robot given by the 2D

SLAM. We thus obtain a 3D point cloud representing partially the environment according to the movements of the robot that is used to categorize the various types of obstacles and to assist the target detection. We use the PCL (Point Cloud Perception) library to characterize the structure of the environment (corridors, walls,…) by selecting a relevant volume, filtering the noise, reducing the density of the points used and extracting the planes corresponding to walls with RANSAC algorithm.

An illustration of the algorithm used to detect horizontal planes and extract objects is shown on Figure 7.

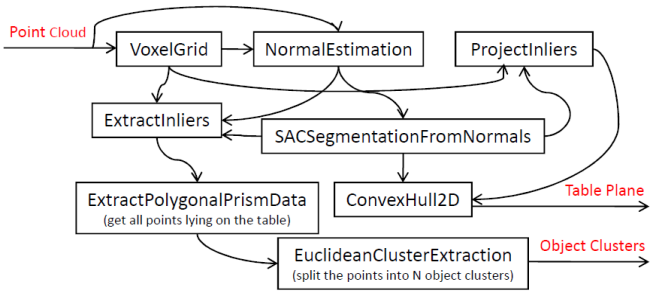

Figure 7. Illustration of table detection and object extraction with PCL library

V. EXPERIMENTAL RESULTS

For the experimental part, we carried out tests in an indoor environment of an approximate surface of 61,2 m^2, containing 3 rooms and a corridor. Our robot needed about 733s to explore the environment and come back to the starting point. During the mission, the Pan-Tilt camera took about 1400 images during the movement and 10 images at each exploration point.

The final result corresponding to the semantic map obtained at the end of the run is shown on Figure 8. Green color corresponds a correct detection with a correct label. Blue color corresponds to a correct detection with a false label. Red color refers to a false detection and pink color refers to missed objects. We note that the missed objects are small objects that are located in the corner of the rooms. Some statistics related to the detection module are shown below.

correct detection / correct label	61,55 %
correct detection / false label	7,69 %
false detection	15,38 %
missed objects	15,38 %

To explore the whole environment, the robot needed 20 exploration points which are shown and numbered in Figure 9. The robot trajectory is quite complex because of the visual exploration strategy which tries to discover a big unknown place first. The visual area covered by the camera is shown on Figure 10 (Right) and room detection is shown on Figure 10 (Left).

An interface (see Fig.11) displaying the map, the robot trajectory, the object detection in 3D and the mission replay has also been developed using the Peekabot library.

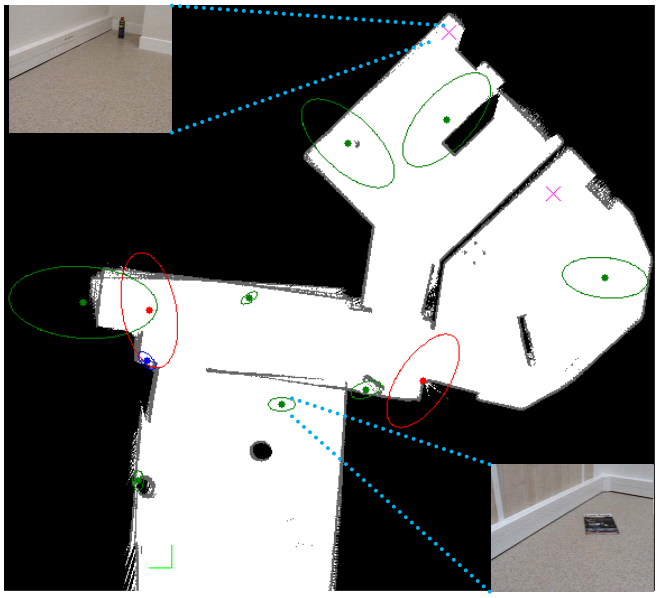

Figure 8. Semantic map showing the environment structure and the objects position with their ellipses of uncertainty. The starting point is shown as a green set of axes. One detected object and one missed object are shown

Figure 9. Exploration map: blue points correspond to exploration points with the robot orientation

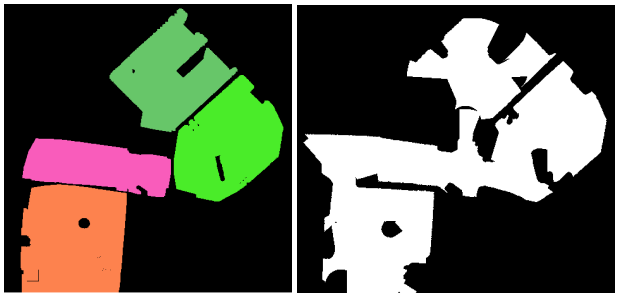

Figure 10. Left: Room detection. Right: view-map showing areas perceived by the camera (91% of the entire surface)

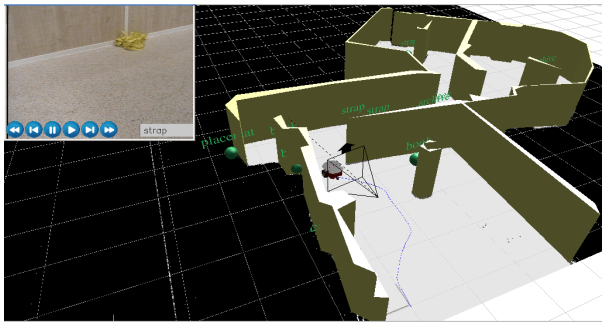

Figure 11. Interface displaying the map and the object detection

VI. CONCLUSION & PERSPECTIVES

In this paper, we presented our solution to participate in the CAROTTE competition. This first year enabled us to design a competitive mechanical system and a robust hardware and middleware platform. Moreover, concerning the software part, we have validated the mapping and the exploration modules. The laser resulting maps are clean and accurate and the area covered by vision is nearly complete. The object detection module gave good results thanks to several approaches of object detection, using color and keypoints. As a result, semantic maps are mainly faithful to the real world. The guidance module revealed some problems relative to the avoidance of some obstacles likes see-through walls. To overcome this default, we are working today to combine effectively laser and sonars data.

For the second year of the challenge, the software robustness will be more deeply evaluated because many new constraints will be added to the competition like a composite ground (carpet, grid, sand, grass, gravel...), more objects (real and virtual) and more complex environments (multi-level, mirror wall...).

We are currently improving the 3D mapping using the vertical laser combined with a high resolution color camera and a time of flight camera, which have the advantage of fast imaging and high lateral resolution combined with the depth information of the captured scene. We are also integrating depth perception in order to improve object recognition.

REFERENCES

[1] R.B. Rusu, N. Blodow, Z. Marton, A. Soos and M. Beetz, "Towards 3D object maps for autonomous household robots," Intelligent Robots and Systems (IROS 2007), pp. 3191-3198, 2007.

[2] A.J. Davison, I.D. Reid, N.D. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," IEEE Trans. PAMI, 2007.

[3] T. Lemaire, S. Lacroix, "Long term SLAM with panoramic vision," Journal of Field Robotics, 24(1): 91-111, 2007.

[4] B. Morisset, R.B. Rusu, A. Sundaresan, K. Hauser, M. Agrawal, J.-C. Latombe and M. Beetz, "Leaving Flatland: Toward Real-Time 3D Navigation," In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, May 2009.

[5] O. Booij, B. Terwijn, Z. Zivkovic and B. Kröse, "Navigation using an appearance based topological map," In proceedings of the IEEE International Conference on Robotics and Automation(ICRA), 2007.

[6] A. Angeli, D. Filliat, S. Doncieux and J.-A. Meyer, "A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words," IEEE Transactions On Robotics, Special Issue on Visual SLAM, 2008.

[7] A. Tapus, N. Tomatis and R. Siegwart, "Topological Global Localization and Mapping with Fingerprint and Uncertainty," In Proceedings of the International Symposium on Experimental Robotics (ISER), Singapore, 2004.

[8] O.-M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann and W. Burgard, "Supervised semantic labeling of places using information extracted from laser and vision sensor data," Robotics and Autonomous Systems Journal, 55(5):391-402, 2007.

[9] D.F. Wolf and G.S. Sukhatme, "Semantic Mapping Using Mobile Robots," IEEE Transactions on Robotics, vol. 24, Issue 2, pp. .245– 258, April 2008.

[10] A. Torralba, K. Murphy, W. Freeman and M. Rubin, "Context-based vision system for place and object recognition," ICCV'03 (Intl. Conf. on Computer Vision), 2003.

[11] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[12] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding (CVIU), vol. 110, No. 3, pp. 346-359, 2008.

[13] S. Hinterstoisser, S. Benhimane, N. Navab, P. Fua and V. Lepetit, "Online Learning of Patch Perspective Rectification for Efficient Object Detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska (USA), 2008.

[14] G. Csurka, C. Dance, L. Fan, J. Williamowski and C. Bray, "Visual categorisation with bags of keypoints," ECCV04 workshop on Statistical Learning in Computer Vision, pp. 59–74, 2004.

[15] B. Yamauchi, "Frontier-based exploration using multiple robots," In Proceedings of the Second international Conference on Autonomous Agents, Minneapolis, Minnesota, United States, May 1998.

[16] B. Tovar, L. Muñoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy and S. Hutchinson, "Planning Exploration Strategies for Simultaneous Localization and Mapping," Journal on Robotics and Autonomous Systems, vol. 54(4), pp. 314-331, 2006.

[17] T.C. Shermer, "Recent results in art galleries," In Proceedings of the IEEE, pp. 1384–1399, 1992.

[18] J.K. Tsotsos and K. Shubina, "Attention and Visual Search : Active Robotic Vision Systems that Search," The 5th International Conference on Computer Vision Systems, 2007.

[19] K. Sjö, D.G. Lopez, C. Paul, P. Jensfelt and D. Kragic, "Object Search and Localization for an Indoor Mobile Robot," Journal of Computing and Information Technology - CIT 17, vol.1, pp. 67-80, 2009.

[20] H. Gonzalez-Banos and J. Latombe, "A randomized Art-Gallery Algorithm for Sensor Placement," In Annual Symposium on Computational Geometry, 2001.

[21] B. Yamauchi, "A frontier-based approach for autonomous exploration," In proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997.

[22] N.J. Butko, L. Zhang, G.W. Cottrel and J.R. Movellan, "Visual Saliency Model for Robot Cameras," In International Conference on Robotics and Automation, 2008.

[23] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," In Proceedings of the International Conference on Robotics and Automation (ICRA), 2007.

[24] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," In Proceedings of the International Conference on Computer Vision, 2003.

[25] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2006.

[26] R.C. Bolles and M.A. Fischler, "A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data," IJCAI 1981: 637-643, 1981.

[27] R.C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," In The International Journal of Robotics Research 5 (4): 56–68, 1986.

An integrated system for teaching new visually grounded words to a robot for non-expert users using a mobile device

Pierre Rouanet and Pierre-Yves Oudeyer*
Flowers Team
INRIA Bordeaux Sud-Ouest
351, cours de la Libération
33405 TALENCE - France
pierre.rouanet@inria.fr

David Filliat**
UEI - ENSTA ParisTech
32 boulevard Victor
75015 PARIS - France
david.filliat@ensta.fr

***Abstract*— In this paper, we present a system allowing non-expert users to teach new words to their robot. In opposition to most of existing works in this area which focus on the associated visual perception and machine learning challenges, we choose to focus on the HRI challenges with the aim to show that it may improve the learning quality. We argue that by using mediator objects and in particular a handheld device, we can develop a human-robot interface which is not only intuitive and entertaining but will also "help" the user to provide "good" learning examples to the robot and thus will improve the efficiency of the whole learning system. The perceptual and machine learning parts of this system rely on an incremental version of visual bag-of-words. We also propose a system called ASMAT that makes it possible for the robot to incrementally build a model of a novel unknown object by simultaneously modelling and tracking it. We report experiments demonstrating the fast acquisition of robust object models using this approach.**

I. INTRODUCTION

Social robots are drawing an increasing amount of interest both in scientific and economic communities. These robots should typically be able to interact naturally and intuitively with non-expert humans, particularly in the context of domestic services or entertainment. Yet, an important challenge needs to be addressed: providing the robot with the capability to adapt and operate in uncontrolled, novel and/or changing environments, in particular when interacting with non-expert humans. Thus, the robot should have the ability to discover its environment. Among the various challenges that this implies, we focus here on the problem of how a robot can learn through the interactions with the human and in particular, how a non-expert human can teach a new word to a robot typically associated with an object in its close environment.

However, in its full generality, this brings up very hard problems and in particular the issue of how the robot can infer the conceptual meaning of a new word [1]. Here, we will restrict ourselves to the specific case where a given word is only associated with a single concrete object. Thus, we are not dealing with concepts, but only with visual object instance recognition. Nevertheless, this is a very ambitious project since several important obstacles still need to be crossed:

- **Attention drawing:** How can a human smoothly, robustly and intuitively draw the attention of a robot towards himself and towards the interaction when the robot is doing its own activity? How can the human draw the robot's attention even if he is not in its field of view?
- **Pointing:** How can a human designate an object to a robot and draw its attention toward this particular object? If the object is not in the field of view of the robot, how to push the robot to move adequately? When the object is within the field of view, how can the human point at this particular object and how could the object be robustly extracted from its background?
- **Joint attention:** How can the human understand what the robot is paying attention to? How can the human accurately know what the robot is seeing? How can joint attention be realized [2][3]?
- **Naming:** How can the human introduce a symbolic form that the robot can perceive, register, associate with the object, and later on recognize when repeated by the human? What modaliti(es) shall be used to ensure ease of use, naturalness, and robustness?
- **Categorization:** How can associations between words and visual representations of objects be memorized and reused later on to allow the human to have the robot search an object associated with a word he has already taught to the robot? Like when human children learn language, social partners can only try to guide the acquisition of meanings but cannot program directly the appropriate representations in the learner's brain. Thus, the process of data collection may lead to inappropriate learning examples. False interpretations could ensue from a wrong data collection. How can we maximize the efficiency of example collection while keeping intuitive and pleasant interaction with non-expert humans? How can we recognize when two examples are related to the same object? Could the human help the robot during this process?
- **Searching:** How can a human intuitively ask for the robot to find or point to an already known object? How can easily and robustly the matching word can be recognized? How can the user intuitively help the recognition?

Thus, in order to give the ability to a non-expert human to

teach new words to its robot, we have to address *visual recognition, machine learning* and also *Human-Robot Interaction (HRI)* challenges. In this paper, we argue that by focusing on the HRI challenges we could significantly improve the whole learning system. We therefore propose a system to tackle some of these challenges (attention drawing, pointing, joint attention and naming) and illustrate the improvement in the efficiency of the learning system brought by this approach.

II. Related Work

Over the past few years, some works tried to address these issues by transposing human-human modes of interaction. Scassellati developed mechanism of shared attention through gaze monitoring and pointing gesture [4]. In his work, he used a fixed upper-torso humanoid robot. Many researches also tried to recognize pointing gestures in order to designate objects to a robot [5][6]. Steels et al. developed a complete social framework based on direct interactions (pointing gestures and speech recognition) to allow an user to teach words to an AIBO [1]. In this work, the authors are more specifically focusing on the visual perception and machine learning issues. Yet, the HRI was identified has a major limitation of their system. In particular, they showed that the lack of robustness of the interface lead to some bad learning examples and so decreased the learning performance [7]. Thus, in this paper we are proposing an integrated system allowing to teach new words to a robot, with a special attention on the HRI challenges. More specifically, we tried to develop an intuitive, efficient and entertaining interface, which also makes it possible for the user to provide the system with good learning examples. By doing so, we are hoping to improve the performance of the whole learning system.

As presented above, in most of the related works, authors choose to use the direct interactions (gesture recognition, gaze tracking or voice recognition) to try to address the above mentioned HRI challenges. In particular, this approach potentially provides really natural interactions which is particularly important with non-expert users. Unfortunately, existing associated techniques are not robust enough in uncontrolled environments (due to noise, lighting or occlusion) and most social robots have a body whose shape and perceptual apparatus is not compatible with those modes of interaction (small angle of view, small height...). This implies that such an approach will fail if one is interested in intuitive and robust interaction with non-expert users in unconstrained environments.

We argue that one way to help to achieve intuitively and robustly some of the functionalities presented above is to develop simple artefacts that will serve as mediators between the human and the robot to enable natural communication, in much the same way as icon based artefacts were developed for leveraging natural linguistic communication between human and certain bonobos [8]. More particularly, we argue that using mobile devices, such as illustrated in figure 1 may enable to circumvent some of theses problems. Though it may seem less natural to use a device as a mediator

Fig. 1. Using a handheld device as a mediator object to control the movements of a social robot.

object between humans and robots, by allowing a robust, reliable and working interaction, it may lead to actually more practical and usable interactions. Such interfaces may provide pleasant and nonrestrictive interactions, and so rather quickly become sort of "natural" interactions.

These kinds of interfaces have already been used to interact with a robot. Kemp. et al have shown how a laser pointer can be intuitively used by people with severe motor impairments to robustly designate objects to a robot [9]. Thanks to the laser spot light, the human can also accurately know what he is pointing at. Yanco et al. used an interface based on an input device (touch screen or joystick) to select objects which will be grasped by a wheelchair mounted robotic arm [10]. In their work the user can directly monitor the object selection on the screen of the device. As we try to do in our system, they both can draw the robot attention toward objects and so realize joint attention between the human and the robot. However their robot is able to automatically grasp the object from a detected 3D spot, in a framework that requires image segmentation algorithm and/or a priori objects knowledge. If objects are not known beforehand these are still hard problems.

In order to circumvent this problem, we argue in this paper that is possible to have the user segmenting himself the object from the image in an intuitive manner by using a handheld touch-screen device. Indeed, the screen of the device can be used to provide the human with information about what the robot is perceiving, which is interesting with non-expert users who are particularly proned to make assumptions about the capacity and behavior of the robot. But it also allows to transfer information from the human to the robot, through easily perceivable gestures [11]. Moreover, these sketches and gestures are natural cues and so are natural for people to use [12]. Thus, we can develop intuitive collaborative interaction allowing the human to supervise the robot and allowing the robot to take advantage from the human capabilities [13]. In particular, here we can display the camera stream on the screen and let the user to encircle the interesting object on the touch-screen. Finally, handheld devices allow the human

to be next to the robot and physically engaged, for example allowing to catch objects and waving them physically in the robot's field of view. They also allow tele-interaction with the robot through the video feedback of the camera.

Other handheld device based interfaces have been developed recently. For instance, Fong et al. used a PDA for remote driving [14], and Kaymaz et al. used it to tele-operate a mobile robot [15]. Sakamoto et al. showed how they can control a house cleaning robot through sketches on a Tablet PC [16]. However, to our knowledge nobody used a handheld device for such teaching interactions. We already proposed a prototype based on a handheld device to teach new words to a robot [17]. This prototype was developed with a special care to the classical design lessons in HRI and HCI [18][19][20]. The exploratory study indicated that it was a promising approach, providing an intuitive and efficient interface for non-expert users. We also compared different interfaces for showing object to a robot and concluded that the gesture interface based on a handheld device was stated as the most satisfying by the users [21]. In this paper, we propose a full-system with advanced visual perception, machine learning and HRI components.

III. Outline of the system

A. Visual perception

We adopted the popular "bags of visual words" approach to process images in our system. Bags of visual words is a method developed for image categorization [22] that relies on a representation of images as a set of unordered elementary visual features (the words) taken from a dictionary (or code book). Using a given dictionary, a classifier is simply based on the frequencies of the words in an image, thus ignoring any global image structure. The term "bag of words" refers to document classification techniques that inspired these approaches where documents are considered as unordered sets of words. Several applications also exist for robotics, notably for navigation (e.g. [23], [24]).

The words used in image processing are based on local image features such as the SURF keypoints [25] we are using in this paper. As these features can be noisy and are represented in high dimension spaces, they are categorized using vector quantization techniques (such as k-means) and the output clusters of this categorization are the words of the dictionary. Instead of building the dictionary off-line on an image database as is performed in most applications, we use an incremental dictionary construction ([26]) that makes it possible to start with an empty dictionary and build it as the robot discovers its surroundings. This make it possible to learn objects without any a priori on the object type or the environment of the robot.

This model has two interesting characteristics for our application: the use of feature sets make it robust to partial object occlusions and the feature space quantization bring robustness to image noise linked to camera noise or varying illumination. More over, with the incremental dictionary construction, this quantization is adapted as the environment evolve (for example when light change from natural to artificial).

B. Machine learning

For our application, the classifier designed for object recognition should be trained incrementally, i.e. it should be able to process new examples and learn new objects without the need to reprocess all the previous data. To achieve that, we use a generative method in which training entails updating a statistical model of objects, and classifying entails evaluating the likelihood of each object given a new image.

More specifically, we use a voting method based on the statistics of visual words appearance in each object. The recorded statistics (according to the learning method described later) are the number of appearance a_{wo} of each visual word w of the dictionary in the training examples of each object o. For object detection in a new image, we extract all the visual words from this image and make each word w vote for all objects o for which $a_{wo} \neq 0$. The vote is performed using the *term frequency–inverted document frequency (tf–idf)* weighting [27] in order to penalize the more common visual words. An object is recognized if the quality of the vote result (measured as the difference between the best vote and the second best) is other a threshold.

Estimating the statistics a_{wo} require the labelling of examples with the associated object name. As will be described later, we will use two methods for example labelling depending on the information given by the user : labelling the whole image or labelling only an image area given by the user and representing the object . Additionally, we will propose a new scheme for automatically labelling new examples from an initial user labelled example (see section III-D).

C. Human Robot Interaction

We choose to use the Nao robot as our test platform. Indeed, to us it well represents the present of social robotics: with a toy-aspect and classical inputs (camera, microphone). Furthermore, it is a biped robot and it has a complex skeletal so it leads to complex motions. Finally, as it is a humanoid, user will probably be more proned to teach it new words.

Our system was embedded on an Apple iPhone used as a mediator object between the human and the robot. We choose this device because it allows to display information on the screen to the user and also allows to interact through "natural" gestures through a large amount of possibilities due to the multi-touch capacities. Moreover, thanks to the large success of the iPhone we can take advantage of a well-known interface, allowing ease of use.

In this system, the screen of the handheld device displays the video stream of the robot's camera (at about 15 fps). It accurately shows what the robot is looking at, which can thus be monitored by the user allowing to resolve the ambiguity of what the robot is really seeing (see figure 2). As mentioned above, achieving such an ability with direct interaction is difficult with personal robots such as Nao humanoid or the AIBO robot due to the specific morphology of such robots

Fig. 2. We display the video stream of the camera of the robot on the screen. This allows to accurately monitor what the robot is seeing and thus really achieving joint attention situation.

and the particular characteristic of their camera, in particular with non-expert users who are specifically proned to make really ambitious assumptions about the robot capacity. Moreover, having a visual feedback seems to entertain the user [21], while the robot is moving for instance.

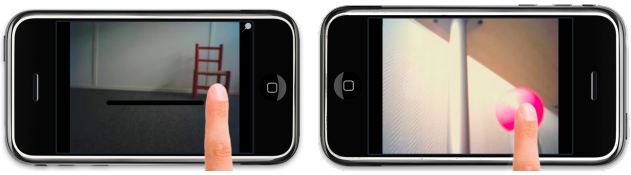

Fig. 3. Drawing attention towards an object: the user first sketches directions to position the robot such that the object is in its field of view (left), and if he wants to center the robot's sight on a specific spot just tap on it (right).

When the human wants to draw the robot attention toward an object which is not in its field of view, the user can sketch on the screen to make it move in an appropriate position: vertical strokes for forward/backward movements and horizontal strokes for right/left turns. Elementary heuristics are used to recognize these straight sketches. The moves of the robot are continuous until the user re-touch the screen in order to stop it. Another stroke can directly be drawn to go on the next move (for instance, go forward then directly turn right). Pointing on a particular point on the screen makes the robot look at the corresponding spot (see figure 3).

When the user wants to show an object which is in the field of view of the robot, and thus on the screen, in order to teach a name for this object, it sketches a circle around this object on the touch screen (as shown on figure 4). Circling is a really intuitive gesture because users directly "select" what they want to draw attention to. This gesture is particularly well-suited to touch-screen based interactions. For instance, Schmalstieg et al. used the circling metaphor to select objects in a virtual world [28]. As for the straight strokes, heuristics are here used to recognize circular sketches, based on the shape of the stroke and the distance between the first and the last point of the sketch. This simple gestures has two important functions:

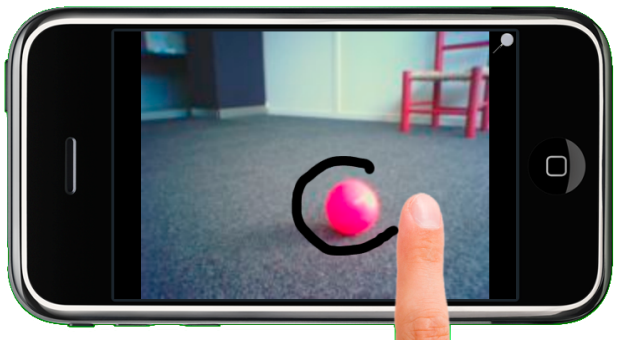

Fig. 4. Encircling an object allows the user to notify the robot that he wants to teach a name for this object. But it also provides an useful rough object segmentation.

- First, it allows to clearly separate, by using two different gestures, the action of drawing the robot attention toward an object and the user's will of teaching a new word for this object.
- Second, circling is also a crucial help for the robot since it provides a rough visual segmentation of the object, which is otherwise a very hard task in unconstrained environments. With the stroke and the background image, we can extract the selected area and define it as our object's image. Classical computer graphics algorithms are used to compute this area (Bresenham line drawing and flood fill).

Fig. 5. Some objects can not be segmented with the classical object segmentation algorithms. For instance, on the left the object has almost the same color than the background. In the middle example, the object is not movable and so can not be segmented with the motion based segmentation method. On the right example, an automatic method can not guess if the user wants to show only the head of the giraffe or the whole poster. Furthermore, this is a 2D object and so the range method can not deal with it.

In this paper, we argue that object segmentation is still a hard task in unconstrained and unknown environments. Different approaches have been developed over the past years to address this problem. However, they are still suffering from a lack of robustness. For example, *Region growing algorithms* try to address this problem by determining regions where color or texture are homogeneous. These region are iteratively expanded from a seed [29]. Yet, these algorithms can not deal with complex objects made up of several subparts with various colors and textures. Moreover the colors or textures of the object can also be similar to the background

(examples are shown on the figure 5). A lot of researches have also studied how the boundaries of the object could be determined through its motion : *motion based segmentation* [30]. Although, this algorithm can only segment carryable or movable object. *Range segmentation* algorithms use images containing depth information to compute regions belonging to the same surface [31]. Of course, this approach does not allow to recognize 2D objects such as a poster. By asking the user to segment the image with a circling stroke, we circumvent all the above mentioned problems and we can deal with all the kind of objects, allowing us to work in unconstrained environments.

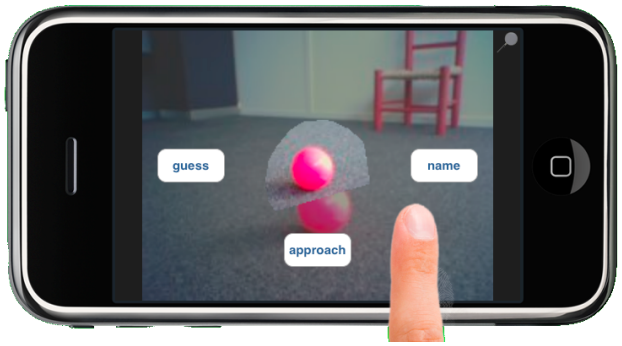

Fig. 6. When a joint attention situation has been achieved on an object, a set of actions are presented to the user, which can in particular decide to teach a name for this object

Once this object is encircled, a menu pops-up showing several interaction options : *naming*, *guessing* or high-level actions such as *approach*, as drawn on figure 6. The "name" choice makes the system wait for the user to enter a referent. In this prototype, we choose to enter the name through a virtual keyboard which allows us to quickly have a whole working system (see figure 7). Moreover, it also allows to circumvent the name recognition problem. Obviously, in further work other modalities such as vocally naming will be used. Once the user has entered the word he wanted to teach, the visual features inside the circle are added to our learning system and the corresponding words are labelled as belonging to the model of the object (as described in section III-B).

Fig. 7. When the user has decided to teach a name for an object, the system is waiting for him to enter a word with the virtual keyboard.

Later on, when the robot has learnt some words, the user can ask it to look for one of this object by entering the looking menu and selecting the object he wants the robot to look for, as shown on figure 8. A simple search algorithm have been developed to move the robot until it detects the searched object by using the visual bag-of-words system.

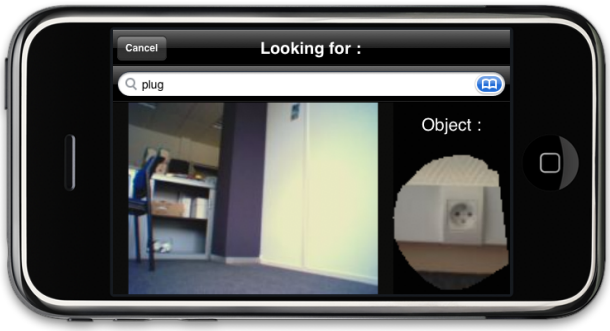

Fig. 8. Once the robot has learnt some words, the user can ask the robot to look for one, by selecting it on the looking menu.

D. Active simultaneous modelling and tracking : ASMAT

As presented above, each time the user encircles an object, our system only improves the model of the object with a single image. However, the appearance of an object can drastically changes from different points of view, so multiple images of an object are needed in order to be able to recognize it, in spite of the point of view [32]. So we develop a system allowing to automatically collect many learning examples, with different points of view, from a single user's intervention (see figure 9). Through this approach, we think we can get a much more robust model of an object and provide a less restrictive interaction to the users.

Fig. 9. We can automatically extract image of an object from different points-of-view with our *active simultaneous modelling and tracking (ASMAT)* system in which the robot turns around the objects, and thus improve the model of the object and make it more robust.

With this system, when the user encircles an object on the screen of the device, we make the robot turn around the object and take pictures of the object from different points of view. However, this a hard problem because we do not have any a priori model of the object in order to track it, and to get a model of the object we need to be able to track it. In order to achieve such an ability we need to be able to simultaneously model and track an object without any prior data. A related system was already presented by Dowson et al. [33], but using previously recorded video, while we can here directly define the movements of the camera. For this reason, we called our system *active simultaneous modelling and tracking (ASMAT)*. With this system, the robot incrementally constructs a model of the object, thanks to the

incremental bag-of-words method which is used at the same time to robustly track the object and thus to enable the robot to turn around it.

The robot incrementally turns around the object through N steps. The robot goes from one position to the next one by moving sideways and turning itself in order to keep the tracked object in the center of its sight. For each stationary position, we lightly move the head of the robot to quickly get M different images of the object. For each image, we first find the SURF keypoints (in order to work in real time) matching the model of the object. We compute the gravity center of these points and compute the average distance to this center. We then filter the points which are too far from this average region. Finally, we define the bounding box of the elected points as the object, i.e. we add to the model of the object all the keypoints inside this box (see algorithm 1).

Such an approach allows the learning system to quickly get a robust model of the object (as shown in the experiments in section IV-B). However, this system can lead to exponential deviation due to fact that the constructed model is also used to track the object, thus the more the system is mistaken more he will be mistaken. To circumvent this problem, we could display on the screen of the iPhone the bounding box of the tracked object allowing the user to stop the robot as soon as it goes wrong and for instance, ask the user to re-encircle the object in order to restart the process. However, this possibility has not yet been implemented and thus evaluated.

Algorithm 1 ASMAT(user_encircled_image)

```
keypoints ← extract_keypoints(user_encircled_image)
update_object_model(keypoints)

while not user_stop() and i < N do
  for j in 1 to M do
    move_robot_head()
    keypoints ← extract_keypoints(robot_camera)
    matches ← find_matching_object_model(keypoints)
    elected ← filter_isolate_points(matches)
    bb ← compute_bounding_box_from_points(elected)
    for each kp in keypoints inside bb do
      update_object_model(kp)
    end for
  end for

  walk_step_around_object()
  keypoints ← extract_keypoints(robot_camera)
  matches ← find_matching_object_model(keypoints)
  center ← compute_gravity_center(matches)
  robot_center_sight(center)
  i ← i + 1
end while
```

IV. Experiments

A. Encircling

In order to test our integrated recognition system (visual perception, machine learning and HRI), we needed to recreate a realistic test environment. First of all, we characterize the environment and the kind of object the humans would teach to robot in a home environment. To us, these objects will probably be everyday objects which can be found for instance in a living room. We can then define two main groups of objects:

- small, light and carryable objects as a newspaper, keys or a ball
- bigger, fixed objects as a plant or a plug

We think that the first categories will represent the most important part of taught objects. Furthermore, they also probably are the hardest to recognize due to the background changes. To our knowledge, the most matching database would be the ETH-80 image set. However, this database uses class object and not only instances. Furthermore, a neutral background is used which is not representative of unconstrained environments. The point of view is also quite similar from one image to another, while the robot will have to recognize an object from different points of view. Thus, we decided to create our own database with a special attention to our criteria. We chose 20 different everyday objects, which are rather small, carryable and well-textured. As our objects are carryable, we must be able to recognize them in spite of their location, i.e. with different backgrounds. So we chose five different backgrounds (on the ground, on a desk, at a window, in front of a bookcase and in the kitchen) and took two pictures by object and by background. So, finally we got 10 images by object taken with different points of view but at the rather same height (figure 10). Every images has been roughly segmented with a stroke encircling the object. Our database is deliberately rather small because we want to be able to recognize an object with few learning examples provided by the user. To us, ten user's examples seems to already be a maximum in order to keep an nonrestrictive interaction. The image were taken and encircled by the authors. Furthermore, they were taken with the camera of the iPhone, and converted to a low resolution (320x240) to correspond to a common resolution.

Fig. 10. Several examples of a specially constructed objects database corresponding to the kind of objects we think an human would like to teach to its robot (everyday object, small and carryable). The images were taken with different backgrounds in order to be able to recognize objects in spite of their location.

In order to test our recognition system, we use the following experimental protocol.

- We randomly choose N images per object.
- We train our learning system with these images.
- We test our learning system with the other images (10 - N images per object).
- The test is repeated 20 times in order to circumvent the randomize effect.
- The final results are the mean recognition rate of each test.

As shown on the figure 11, encircling the objects allows us to improve the recognition rate by 20% in average. So, we can see that we the recognition rate maximum (about 80%) was reached by giving six encircled learning examples, while the maximum with nine full images was not reached. By simply encircling the objects on the screen, the user can improve our recognition system and in particular can achieve robust recognition with fewer learning examples. Thus, we can reduce the number of user's interventions.

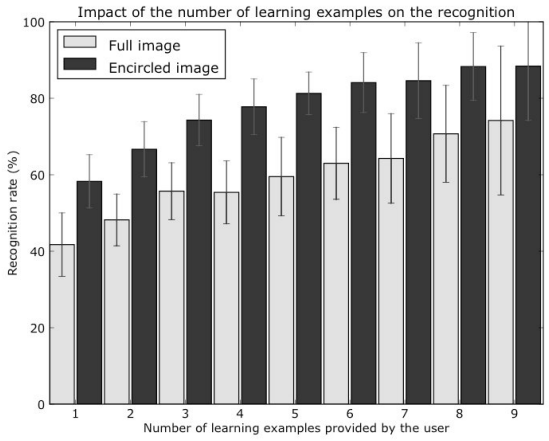

Fig. 11. We can notice that with the encircled images, we have a recognition rate superior of 20% in average than with the full image. Thus, we reach the maximum recognition rate faster (from the sixth encircled example with encircled image while it is still not reached with nine full learning examples).

However, our images were here not gathered in "real" conditions. Indeed, the images were not taken with a robot by non-expert users. Nevertheless, we try to reproduce a plausible interaction, by taking pictures with possible angles of view, with a common resolution for camera... Furthermore, the encircling was done by the authors. Thus, we should, in a future experience, test if non-expert users would provide as good inputs, as were given by the expert users and see if we can get the same results.

B. ASMAT

As mentioned above, we developed a system to automatically get a larger set of learning examples by making the robot turn around the objects. We try to evaluate the impact of such a method on the learning process. So, we designed an experiment where a user taught four different objects to the robot by encircling them on the screen of an iPhone. Then, the robot automatically turned around the object. At each step, the robot moves sideways and forward. Then it turns in order to recenter its sight on the center of the tracked objects. Then we take five snapshots with lightly different positions of the head. We repeat this operation five times, so 25 images were taken by learning example. For each object, the user give five different learning examples with different points of view of the object. We define two conditions:

- In condition A, we only use the first image (the one encircled by the user) to train our recognition system.
- In condition B, we use all the 25 images labelled using our ASMAT system to train the recognition system.

We then used a similar database as the one used above (5 backgrounds, 2 images per background and per object), with our four objects to test the quality of the learning.

As we can see on the figure 12, with the condition A, we have a linear progression of the recognition rate according to the number of learning examples : with five learning examples we reached about 60% of recognition. With the condition B, we can notice a really fast increase of the recognition rate. A maximum (about 80%) is reached from the second learning examples. We can also notice that this maximum is not reached with the condition A even after the fifth examples. Thus, the *ASMAT* system seems to allow the getting of a robust and reliable model of an object with really few user's interventions.

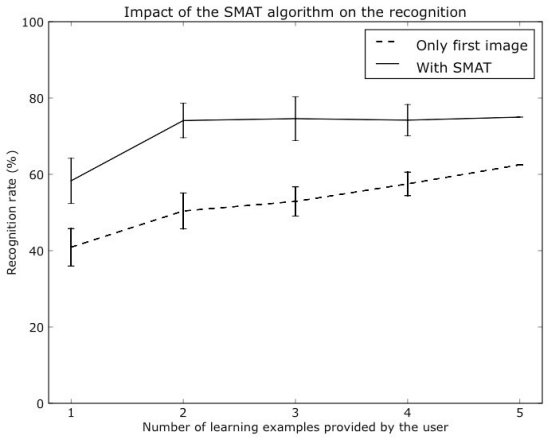

Fig. 12. Recognition rate according to the number of examples given by the user. We define two conditions, one with the *ASMAT*, the other without it. We can notice that by using this system, we can get a more accurate model of an object with fewer user's learning examples.

V. Conclusion and Future Works

A. Conclusion

Our proposed system, based on a combination of advanced HRI, visual perception and machine learning methods, allows non-expert users to intuitively and robustly teach new words to their robot. By using the touch-screen to transfer information, we have developed collaborative interactions, improving

the mutual understanding between the robot and the human. We also showed that the interface may help the user to provide good learning examples which will thus improve the whole learning system.

B. Future works

In future works we will try to evaluate the impact of different interfaces on the learning by designing a comparative user's study with different kind of interfaces (with an iPhone, with a laser pointer and with direct interactions). We will compare them with a "learning quality" measure, but also with satisfaction questionnaires to assess their usability. It would also be interesting to evaluate the impact of the ASMAT system on the user's experience, especially with non-expert users. Thus, it could enhance the interaction, by making it more lively and more entertaining for the users. On the other hand, the extra time taken to do the active learning, may fatigue users.

Then, we will use a vocal naming system and so we will have to be able to compare two vocal words. We will also allow the user to improve the learning through collaborative interactions, where he could help the clustering of the different learning examples, and try to evaluate the real impact of such a feature.

VI. Acknowledgments

The authors would like to thank Jérôme Béchu for his implication in the development and the realization of the different parts of the system.

References

[1] L. Steels and F. Kaplan, "Aibo's first words: The social learning of language and meaning," *Evolution of Communication*, vol. 4, no. 1, pp. 3–32, 2000. [Online]. Available: http://www3.isrl.uiuc.edu/ junwang4/langev/localcopy/pdf/steels02aiboFirst.pdf

[2] C. Breazeal and B. Scassellati, "Infant-like social interactions between a robot and a human caregiver," *Adapt. Behav.*, vol. 8, no. 1, pp. 49–74, 2000.

[3] F. Kaplan and V. Hafner, "The challenges of joint attention," *Proceedings of the 4th International Workshop on Epigenetic Robotics*, 2004.

[4] B. Scassellati, "Mechanisms of shared attention for a humanoid robot," in *Embodied Cognition and Action: Papers from the 1996 AAAI Fall Symposium*, 1996.

[5] K. Nickel and R. Stiefelhagen, "Real-time recognition of 3d-pointing gestures for human-machine-interaction," in *International Workshop on Human-Computer Interaction HCI 2004, May 2004, Prague (in conjunction with ECCV 2004)*, 2004.

[6] V. V. Hafner and F. Kaplan, "Learning to interpret pointing gestures: Experiments with four-legged autonomous robots," in *in Proceedings of the KI2004 Workshop on Neurobotics*. Springer, 2004, pp. 225–234.

[7] F. Kaplan, *Les machines apprivoisées comprendre les robots de loisir*. vuibert, 2005.

[8] S. S. Rumbaugh and R. Lewin, *Kanzi : The Ape at the Brink of the Human Mind*. Wiley, September 1996.

[9] Y. S. Choi, C. D. Anderson, J. D. Glass, and C. C. Kemp, "Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired," in *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. New York, NY, USA: ACM, 2008, pp. 225–232.

[10] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Development and evaluation of a flexible interface for a wheelchair mounted robotic arm," in *HRI '08: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*. New York, NY, USA: ACM, 2008, pp. 105–112.

[11] M. Skubic, S. Blisard, A. Carle, and P. Matsakis, "Hand-drawn maps for robot navigation," in *AAAI Spring Symposium, Sketch Understanding Session, March, 2002.*, 2002.

[12] M. Goodrich and J. Olsen, D.R., "Seven principles of efficient human robot interaction," *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 4, pp. 3942–3948 vol.4, Oct. 2003.

[13] T. Fong, N. Cabrol, C. Thorpe, and C. Baur, "A personal user interface for collaborative human-robot exploration," in *6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS)*, Montreal, Canada, June 2001. [Online]. Available: citeseer.ist.psu.edu/fong01personal.html

[14] T. W. Fong, C. Thorpe, and B. Glass, "Pdadriver: A handheld system for remote driving," in *IEEE International Conference on Advanced Robotics 2003*. IEEE, July 2003.

[15] H. Kaymaz, K. Julie, A. Adams, and K. Kawamura, "Pda-based human-robotic interface," in *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics: The Hague, Netherlands, 10-13 October 2004*, 2003.

[16] D. Sakamoto, K. Honda, M. Inami, and T. Igarashi, "Sketch and run: a stroke-based interface for home robots," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. New York, NY, USA: ACM, 2009, pp. 197–200.

[17] P. Rouanet and P.-Y. Oudeyer, "Exploring the use of a handheld device in language teaching human-robot interaction," in *Proceedings of the AISB 2009 Workshop : New Frontiers in Human-Robot Interaction*, 2009.

[18] J. A. Adams, "Critical considerations for human-robot interface development," in *AAAI Fall Symposium on Human-Robot Interaction*, Cape Cod, MA, November 2002.

[19] J. Drury, J. Scholtz, and H. Yanco, "Awareness in human-robot interactions," *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 1, pp. 912–918 vol.1, Oct. 2003.

[20] H. A. Yanco, J. L. Drury, and J. Scholtz, "Beyond usability evaluation: analysis of human-robot interaction at a major robotics competition," *Hum.-Comput. Interact.*, vol. 19, no. 1, pp. 117–149, 2004.

[21] P. Rouanet, J. Béchu, and P.-Y. Oudeyer, "A comparison of three interfaces using handheld devices to intuitively drive and show objects to a social robot : the impact of underlying metaphors," *RO-MAN*, 2009.

[22] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV04 workshop on Statistical Learning in Computer Vision*, 2004, pp. 59–74.

[23] J. Wang, R. Cipolla, and H. Zha, "Vision-based global localization using a visual vocabulary," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[24] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Real-time visual loop-closure detection," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2008.

[25] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[26] D. Filliat, "Interactive learning of visual topological navigation," in *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008.

[27] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[28] D. Schmalstieg, L. M. Encarnaç ao, and Z. Szalavári, "Using transparent props for interaction with the virtual table," in *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*. New York, NY, USA: ACM, 1999, pp. 147–153.

[29] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.

[30] A. Arsenio, P. Fitzpatrick, C. C. Kemp, and G. Metta, "The whole world in your hand: Active and interactive segmentation," pp. 49–56, 2003. [Online]. Available: http://cogprints.org/3329/

[31] A. Bab-Hadiashar and N. Gheissari, "Range image segmentation using surface selection criterion," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2006–2018, July 2006.

[32] P. Robbel, "Exploiting object dynamics for recognition and control," Ph.D. dissertation, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences., 2007.

[33] N. D. H. Dowson and R. Bowden, "Simultaneous modeling and tracking (smat) of feature sets," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 99–105, 2005.

A bag-of-features framework for incremental learning of speech invariants in unsegmented audio streams

Olivier Mangin
École polytechnique / INRIA, France

Pierre-Yves Oudeyer
INRIA, France

David Filliat
ENSTA ParisTech, UEI, France

Abstract

We introduce a computational framework that allows a machine to bootstrap flexible autonomous learning of speech recognition skills. Technically, this framework shall enable a robot to incrementally learn to recognize speech invariants from unsegmented audio streams and with no prior knowledge of phonetics. To achieve this, we import the bag-of-words/bag-of-features approach from recent research in computer vision, and adapt it to incremental developmental speech processing. We evaluate an implementation of this framework on a complex speech database.

1. Introduction

1.1 Constraints, difficulties and learning scenarios

The goal of this article is to introduce a computational framework that allows a machine to bootstrap flexible autonomous learning of speech recognition skills. Technically, this framework shall enable a robot to incrementally learn speech invariants from unsegmented audio streams and with no prior knowledge of phonetics (i.e. without re-using for example a probabilistic phone recognizer). This requires to bring together various conventional speech recognition techniques with the constraints of developmental learning.

This goal implies several important challenges and issues. First of all, in order to achieve **developmental and cognitive plausibility**, as for example discussed by Brent, (Brent, 1999), our system must be incremental and self-organised. Furthermore it must start with little and generic knowledge on the environment it has to explore, i.e. in our case no knowledge of phonetics, phonotactics or lexicons. Another important issue is the design of interaction channels between the system and its environment, that is to say the kind of **(un-)supervision** it is (un-)exposed to and the **multimodality** of the input channels, as for example coupling speech learning with visual information. Finally the integration of motor channels in the learning process connects learning and action. This introduces issues ranging from goal oriented analysis of the audio stream, to studies of more intrinsic structures of this stream.

Choosing a standpoint between these aspects defines a learning scenario. A broad spectrum of such experimental setups have been developed in previous work and shows how they influence the choice of methods and algorithms. Organizing those scenarios through the degree of supervision they include gives a good way to identify categories of machine learning methods they integrate. First, systems with no interaction with outside word, thus unsupervised, extract intrinsic structure of the speech flow. Such scenarios lead to a kind of "autistic" systems: as they do not share any convention with the outside world, or other sensorimotor channels, no real communication is possible (Park and Glass, 2008). Anyway, the mechanism used in those scenarios are of real interest, for example to be used for bootstrapping (Iwahashi, 2003, Brandl et al., 2008), or as intermediate modules of larger architectures. The second category of scenarios is the one for which supervision takes the form of a reward, thus leading to reinforcement learning frameworks. It is for example the case when Gorin et al. (Gorin et al., 1994) train their system to a specific task: the *How may I help you ?* problem. Finally, the most common scenarios take the form of a completely supervised problem, with for example labeled data. Those labels may be semantic tags (Gorin et al., 1999, ten Bosch et al., 2008) or language tags for language recognition (Ma and Li, 2005). Self-labeled data are also possible, for example, when dealing with multiple sensory-channels.

1.2 Existing architecture for speech learning

We have just seen that the methods and algorithms to be used are quite influenced by these learning sce-

narios. Those methods must be integrated in some learning architecture, designed to fit the requirement of a specific scenario. Those are quite different but we may extract some key principles of their design, some of which are essential in the framework we present in this article.

Sub-lexical and lexical classification These are the abilities to classify sounds into groups that share similarities, such as phonemes, and to build a representation of words on top of those groups of basic sounds, which includes the compliance to phonotactic rules. To achieve lexical and sub-lexical classification, two heuristics have to be used: topological information, representing the proximity between sounds, and statistical information used to assimilate different sounds, that have the same function in the language. It has indeed been shown (Kuhl, 2004) that, as young infants are initially able to distinguish different sounds, this ability disappears for sounds that are different but functionally equivalent in their mother language. This phenomenon is necessary to achieve robustness against the great variability of pronunciation of a given phoneme.

Sub-lexical classification may be achieved by Recurrent Neural Networks, as in (Roy and Pentland, 2002), or Hidden Markov Models as in (Ma and Li, 2005); but in those examples, this classifier is trained offline, before the experiment, in a static manner, as a *universal sound recognizer* which does not fit our developmental requirements. Aimetti (Aimetti, 2009) build a system that learns a lexical representation, computed using similarity measures on segments of speech; one example of those segments is used to represent each keyword. Park and Glass used a graph clustering method to group similar sound segments as sub-lexical entities (Park and Glass, 2008), thus autonomously learning such a representation. The learning of this classification is often treated as a clustering problem.

Segmentation Whereas utterance segmentation is quite easy through silence recognition, word segmentation is indeed a difficult task (even for standard written text when spaces are removed). Many speech recognition systems are based on the ability to find a word segmentation, whereas others recognise utterance without performing such a segmentation. For example *segmental dynamic time warping* methods uses dynamic programming to find similar sound segments between speech examples. Then they define sub-lexical units as those segments (Aimetti, 2009, Park and Glass, 2008, Gajjar et al., 2008). On the other hand non-negative matrix factorization methods, such as in (ten Bosch et al., 2008, ten Bosch et al., 2009) also show great results in speech recognition. This method builds an internal word representation from whole utterances, without being designed on the ability to segment words. It is anyway, afterward, able to find such a segmentation, as a consequence of the recognition process.

Semantic structure Matching a lexical representation of spoken utterances to a more structural representation, which may include grammar extraction, or syntactic analysis, requires specific integration of such structure in the recognition system. Methods designed toward this goal often use a predefined structure, to which the utterances are mapped. For example in Iwahashi's' experiment (Iwahashi, 2003), this semantic consists of (object, action, position) associations and is analysed using a graph structure adapted to this grammar. Without predifined implementation, the system must find an origin for these semantic representations.Other input channels such as the vision channel or motor channels, in the case of action oriented goals, are generally part of this process. It may then be difficult to design a system that autonomously builds its own grammar representation without dealing with the corresponding structures in other sensorimotor channels. For example Gorin et al. (Farrell et al., 1993, Gorin et al., 1994) use multilayer neural networks to map the recognition of some words to an action. In this experiment the semantic of the environment was based on action choices, that is to say on a kind of motor channel.

Memory-like architectures To build larger and more realistic systems, that are capable of long-term learning, it often becomes necessary to work with a model of memory. Actually, having an incremental system often brings growth in data size, leading to memory usage and computation. The ACORNS[1] (Aimetti, 2009, ten Bosch et al., 2008, ten Bosch et al., 2009) project have explored some of these issues by introducing memory levels. The principle is to separate data storage in different levels, where depth in memory is correlated to an increase in organization of the data. This may, to a certain extent, be seen as a compression problem, related to highly organised data in the long term memory, but also introduce an attention mechanism, associated with a short term memory.

2. Applying the bag-of-words method to spoken language invariants discovery and recognition

The main contribution of this article is to adapt the bag-of-words method to a developmental approach of the learning and bootstrapping of speech recognition

[1] **A**cquisition of **Co**mmunication and **R**ecog**n**ition **S**kills, http://www.acorns-project.org

skills. In this approach, the bag-of-words method will be used to bootstrap and maintain incrementally new, potentially multi-scale and multi-type, acoustic representations of speech invariants from unsegmented speech streams and with no prior phonetic knowledge. A lower-level will build those representations in an unsupervised manner, while a higher level of the architecture will consist in re-using those low-level representations to learn to predict a general semantic tag associated to whole utterances. We did not introduce memory handling, but we believe it is a matter of adapting the algorithms from each part of the framework, without changing its global design.

2.1 Background and principle

Bag-of-words methods originate in text classification applications (Joachims, 1997) and have been used with great success in image categorization applications after the seminal work of Sivic and Zisserman, 2003 (Sivic and Zisserman, 2003). The general idea is to represent the text or the image as an unordered collection of local elements chosen in a dictionary (the words in a text and local visual features in an image), thus ignoring the global structure. Using this representation, a classification algorithm can then be used to predict the associated category. In computer vision applications, this representation is very compact thanks to the quantization of local feature representation in the dictionary, while preserving the stable local information and ignoring more unstable global geometry. In most applications, the dictionary is static and requires an initial training phase, but in previous work, we have developed an incremental approach that meet the requirements of developmental systems (Filliat, 2008). We will therefore transpose this method to the speech recognition problem. Yet, for the sake of clarity, we will use the terminology "bag-of-features" instead of "bag-of-words", since the "words" in the bag-of-words approach are not at all equivalent to "linguistic words" in the speech stream and which constitute important speech invariants to be discovered and learnt in our framework.

2.2 Presentation of the framework

Our framework is composed of three distinct layers that we describe below.

- **Continuous Acoustic Feature Vectors (CAF) extraction:** this layer transforms the input audio signal into a set of vectors, each associated with some position information. The goal of this process is to transform the signal into a set of local descriptors, which are more adapted to similarity comparison. Actually the next step requires to have a kind of distance on these vectors, in order to be able to access to a notion of acoustic similarity. This first layer typically uses static sound processing methods (e.g. MFCC or RASTA-PLP, see Section 3.2).

- **Unsupervised clustering:** the role of this layer is to transform each CAF vector from the set obtained above, into a *discretized acoustic feature* (DAF), that is to say a single number. This transformation is accomplished through a clustering process. More precisely this clustering must build incrementally a representation of this DAFs, using the similarity measure inherent to the CAF space. This representation has to both allow retrieval of the DAF corresponding to a given CAF vector and the learning of new DAFs when a CAF vector does not match any known feature.

- **Higher level semantic treatment:** the two previous layers may be seen as a pre-processing, which goal is to transform the input audio signal into a bag of discretized acoustic features, more precisely we get a set of couples, each composed of a DAF and its position. This semantic layer introduces a new representation of the audio signal that allows to efficiently set up higher level statistical treatment, such as keyword recognition or more complex analysis.

Mathematically, this process may be described as following: given an input audio sequence $a \in \mathcal{A}$, a continuous feature vector space $\mathcal{F}$, a set of localization data, such as time position in the utterance, $\mathcal{P}$, a discrete acoustic feature dictionary $\mathcal{D}$:

- extract CAFs: $a \in \mathcal{A} \longrightarrow (v_i, p_i) \in (\mathcal{F} \times \mathcal{P})^\star$
- find corresponding DAFs: $(v_i, p_i) \longrightarrow (f_i, p_i) \in (\mathcal{D} \times \mathcal{P})^\star$

Where, i is a free variable, and for any set $\mathcal{E}$, we call $\mathcal{E}^\star = \bigcup_{k \in \mathbb{N}} \mathcal{E}^k$ the set of finite sequences over $\mathcal{E}$.

In the case of tag inference, the statistical process is then, given a set $\mathcal{T}$ of tags, a mapping: $(\mathcal{D} \times \mathcal{P})^\star \to \mathcal{T}$.

2.3 Modularity and cognitive plausibility

Bag-of-sounds approach has already been used by Ma and Li (Ma and Li, 2005), but with a sub-lexical model built offline and from labelled examples. The novelty of our work is to present a framework based on the ability to learn autonomously a new representation of sound, which enables a completely generic statistical treatment. Indeed the modularity of the framework is present at each of the previous layers.

The first level, whose role is to extract CAFs from input audio sound, may implement a large variety of signal processing treatment on this input stream. It may for example implement windowed spectral or

cepstral analysis of the sound, but may also be composed of more elaborated pieces of information such as pitch or stress patterns. The position labels p_i associated with those vectors may also be of different natures, either just an index in a sequence or a more precise position, and may be completed by other information, such as the width of the input signal relevant to this feature vector. Furthermore, CAFs of different natures may be simultaneously computed, they are then clustered by *distinct* dictionaries, but may be grouped at the end in the *same* bag of DAFs representation. This is a completely transparent way to mix pre-processings of completely different natures.

The second layer may implement any clustering algorithm, since we have a relevant metric on CAFs produced by the previous one. It indeed corresponds to the extraction of acoustic building blocks.

The third layer is completely general as our goal is to provide an intermediate representation for the audio input. We may for example plug onto this representation any classification algorithm, or structured output algorithm.

What we show in the following is that the particular representation we have built retains enough information to enable powerful statistical treatment, and simplifies enough the signal representation to allow such treatment to be efficient. Furthermore, studies on the ability of children to distinguish between sounds seem to indicate that such a representation, even if it is a lot simplified, is a reasonable one for speech recognition (Kuhl, 2004).

The following experiment implements this framework in a very simplified manner: position information is dropped, statistical analysis is reduced to a quite rudimentary scoring method, far from state-of-the-art statistical machine learning approaches. Yet, results will show that this representation is sufficient to predict semantic tags with great accuracy in a large complex database, even with this implementation, thus showing the robustness of the general approach.

3. Specific implementation

3.1 Our experimental scenario

As explained above, in this paper, we adopt a framework where the goal is to allow a robot to progressively learn to predict semantic tag(s) associated to a given speech utterance. For example the robot is incrementally provided with examples of associations between speech utterances and semantic tags, and should accordingly incrementally update its internal representations in order to predict better these semantic tags in new utterances. Semantic tags are technically encoded as keywords referring either to general topic(s) of the utterance, sometimes corresponding to the presence of a particular word in the utterance or to the speaker style or language.

The framework presented in the previous section 2.2 is illustrated by the following steps, in the case of this particular application:

- extract CAFs from the input channel. The CAFs used in our application are described in the next section.
- match these vectors to the dictionary(-ies) and drop the position information, thus creating a bag of DAFs representation of the input signal, and update the dictionary(-ies) if necessary. This particular process is described in section 3.3.
- infer the semantic tag associated with the utterance through a scoring method.

We now present an implementation of this framework and associated experiments. It should be considered as a specific implementation of the general framework that we presented.

3.2 Continuous feature vectors extraction

In our experiment we use Mel-Frequency Cepstral Coefficients (MFCC) and Relative Spectral Transform - Perceptual Linear Prediction (RASTA-PLP) features over a short time window, from Ellis (Ellis, 2005) implementation. The former feature vectors, which are actually time sequences of successive feature vectors, are compared with respect to a Dynamic Time Warping (DTW) distance (Sakoe and Chiba, 1978). This distance takes into account possible insertions and deletions in the feature sequence. It is adapted for sound comparison but does not correspond to an inner product in CAF space, since it is not an euclidean distance, which leads to some new issues.

One other interest of using a DTW distance is to be able to compare sound feature vectors of different length or of varying rhythm. However, in our experiments we used fixed length feature vectors (but the rhythm varies): for each sound utterance we first compute the MFCC sequence corresponding to this audio stream. After extracting this MFCC sequence, we cut it into fixed length features, using a 80 or $150ms$ sliding window. The sliding length used in most of our experiments is one third of the length of the window. However, it is also completely possible to mix several lengths in the same vocabulary or to extract features of random lengths. This may result in a more multiscale-like approach. Those lengths are here around the scale of a phoneme length and give a good trade-off between sufficiently long sequence of MFCC vectors and the DTW quadratic complexity. Furthermore it is relevant to limit this length to get really local descriptors, which we tried

to implement, even if it is not a requirement of the framework.

3.3 Incremental unsupervised clustering

The dictionary must group similar CAF vectors according to the DTW distance into discretized acoustic features. This requires two processes : the dictionary construction and the retrieval of the DAF matching a specific CAF. In our developmental approach we need an incremental dictionary construction able to learn new DAFs, that is to say new vector clusters. We also face a computation time issue for the matching process, thus requiring a dictionary data structure that enables both : efficient algorithms for matching, and the possibility to perform incremental clustering.

Our approach is very similar to the one we used for image processing in (Filliat, 2008). The idea is to represent DAFs by clusters of CAF vectors. Those clusters are organised in a hierarchical way. More precisely, the DAFs are hyperspheres in the continuous feature space, and their centers are organised in a tree structure inspired by the one of Nister and Stewenius (Nister and Stewenius, 2006) where leaves and nodes represent hierarchical clusters. The tree structure is organised as follows:

- each leaf or cluster C is represented by its centroid: a vector v_C,
- each cluster is associated to a hypersphere of radius r_{max} around its centroid. A CAF vector v is therefore part of a cluster C if and only if $d(v, v_C) \leq r_{max}$
- each node of the tree has a limited number of children N_{max} and has an associated centroid n_C which is the mean of its children CAF vectors.

A CAF vector is matched to a cluster by recursively following the child of the node which centroid is the nearest from the searched vector. The dictionary is built by adding these vectors to the tree: we find the nearest cluster; if the vector matches the radius condition regarding to this cluster, it is added inside this one; if not, a new cluster is created initially containing only this vector. This cluster is added as a leaf in the tree, at the same level and with the same father as the previously found nearest cluster. Then we check if the number of children is below N_{max}; if not, the node is split in k nodes, by a k-means process on the centroids of the leaves. The leaves are then distributed to those child nodes. An example of this mechanism, also described by the following pseudo-code, is shown in figure 1.

Algorithm Adding a vector to the cluster tree node

ADD_VECTOR_TO_NODE(*current_node*, *vector*, k, r_{max}, N_{max})

- *current_node* is the node where the vector is to be added,
- *vector* is the vector to add,
- k is the k-means parameter,
- r_{max} is the threshold distance that is used to decide if two vectors are considered identical,
- N_{max} is the maximum number of vectors that a leaf may contain.

```
. if current_node is a leaf
.   let v be the nearest vector in current_node
.   distance (vector, v) ≥ r_max
.   add vector to current_node
.   let n be the number of vectors in current_node
.   if n ≥ N_max
.     new_leaves ← k_means(k, current_node)
.     let new be a new internal node with the
.         elements of new_leaves as children
.     replace current_node by new
.   else
.     add vector to current_node
. else
.   let child be the nearest child from vector
.       in current_node
.   add_vector_to_node(child, vector, k, r_max, N_max)
```

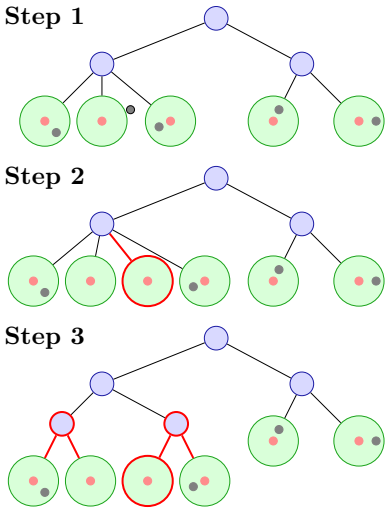

Figure 1: Insertion of a new vector in the hierarchical structure. The nearest leaf is found, but the vector is too far from the center (first step) so a new leaf is created (second step). The new leaf father has now too many children ($N_{max} = 3$) so the node is split in two parts(third step).($k = 2$)

This structure leads to approximate nearest neighbour search, and thus the processes of learning a CAF or retrieving the corresponding DAF are ap-

proximate. Since CAF vectors are themselves noisy, this approximation is naturally handled by the statistical treatment in layer 3. In order to reduce the impact of orientation errors while exploring the tree, which may result in an important final error, for example, if it occurs near the root of the tree, we added the following improvement to the search algorithm.

The idea is to launch more than one search for each request and then select the best results. This is close to *branch-and-bound* techniques and may be implemented in many ways. We tried two implementations of this method. In the first one, for each node reached during the search process, the search is launched again on its b best children, instead of just the best child. By best children we mean the b sons with the lowest distance between their centroid and the requested vector. b is called the backtracking parameter. This method leads to a complexity of $\mathcal{O}\left(n^{\log_k(b)}k\right)$, where n is the number of nodes, k the k-means parameter used to create the tree and b the backtracking parameter. This may be long compared to the $\mathcal{O}\left(k\log(n)\right)$ original complexity.

The second method uses the same idea, but instead of deciding locally which node deserves to be explored, it runs full searches, at the end of which it launches again a search from some node on the tree, where a good candidate path may have been missed. More precisely, during the search, each time a child node is chosen for the proximity of its centroid to the requested vector, its siblings are memorized with some value representing how far they were from the chosen child. When a candidate leaf is finally found, the system is able to reconsider the choices it has made during the search and explore the node which had the best value.

By repeating this process b times, and finally choosing the best candidate nearest neighbor from those found, we are able to minimize the impact of the approximate nature of our structure. The actual complexity of this method is roughly $\mathcal{O}\left(bk\log(n)\right)$.

The second method gave a better trade-off between the number of explored nodes, which corresponds to computation complexity, and the quality of the retrieved approximate nearest neighbor.

3.4 Semantic tag inference

While previous steps were able to build an internal representation for the system, based on topological information, this process had no relation to the final goal of classification. Actually, all the semantics related to the classification task is created in a last step. We used a vote implementation to score DAFs and examples regarding semantic tags.

The idea of the voting scheme is to associate a weight w_i to each DAF i. Let f_i^t be the frequency of DAF i regarding tag t, $f_i^t = \dfrac{n_{i,t}}{n_t}$ where $n_{i,t}$ is the number of co-appearances of DAF i and tag t and n_t the number of appearances of t.

For a query utterance q, where acoustic DAF i appears q_i times, i votes as $V_i = q_i \cdot f_i^t \cdot w_i$, where w_i are weights.

A common way of setting weights w_i is to use a Time Frequency - Inverse Document Frequency (TF-IDF) approach by setting $w_i = log\left(\frac{N_{tags}}{N_{tags}^{(i)}}\right)$, where N_{tags} is the total number of tags and $N_{tags}^{(i)}$ the one of tags whose examples contain DAF i at least once.

From this basis, conditions may be added such as setting all node weights to zero except from leaves, which rely entirely on the a priori chosen size of clusters, that is to say the r_{max} parameter, in our case. One may also choose to allow only nodes near the leaves to have a nonzero weight or to rely entirely on TF-IDF weights. This kind of modifications may bring more scalability and robustness to the system. It also defines which clusters are DAFs: either only leaves or all nodes, and thus the use or not of hierarchical and multi-scale DAFs.

In order to be able to compute this score we store the number of appearances of each DAF in an utterance associated to a particular semantic tag: this corresponds to previously introduced $n_{i,t}$.

The following process is used: while training, for a given utterance with tag t, transformed in a bag of DAFs, for each DAF i, $n_{i,t}$ is increased by one.

During a testing phase, we extract the bag of DAFs corresponding to the utterance. Then, for each tag we compute its score on the utterance, by summing the votes of each DAF. Votes are computed as explained previously, using only the count of co-occurrences, by simple operations over the $(n_{i,t})_{i,t}$ matrix.

4. Results, analysis and further directions

4.1 Databases and protocols

We restricted our work on labeled classification problems, that is to say, sets of utterances associated with a semantic label. Those labels may be words contained in the utterance as well as more general themas, levels of speech, or speakers. The system is trained with such a learning database and then evaluated on its label prediction performance.

During our experiments we worked with two databases. The first one was a home made database in which utterances were single words. This database, which contains twenty three examples of ten different words, was used to evaluate the performances of the nearest neighbor retrieval with word-long features. The second one is a database provided by the ACORNS project, composed of 1000

utterances containing 13 keywords, each spoken by 4 speakers in English adult directed speech; which makes a total of 4000 utterances. An example of sentences used in the database is *Angus is lazy today.* where the semantic tag/keyword is *Angus*.

In the experiments we split the database into a training set and an independent test set to evaluate the system. In order to characterize the efficiency of the learning process as its improvement through training, that is to say the convergence speed of the algorithm, we regularly test the process during the training and visualize its performance at each step.

4.2 Global results

In order to demonstrate the cognitive efficiency of our system we set up the following experiment: for each speaker we randomly split the database in two sets: a train set consisting of 900 examples and a separate test set of 100 examples. The system is trained incrementally with each utterance of the training set; after each 100 train examples, the system is tested on the whole test set. This protocol, which allows us to monitor its progress, is represented in figure 2.

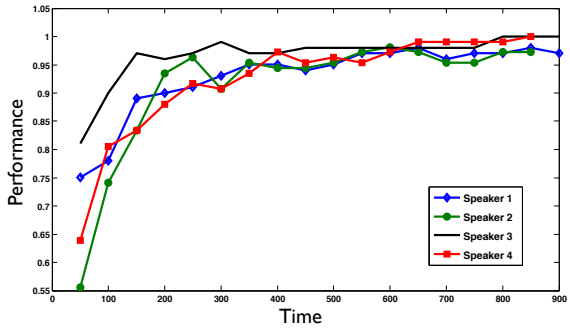

Figure 2: Accuracy against number of training examples for incremental learning and testing, 1000 examples: 900 for training and 100 for testing, one 100 examples test cycle each 50 train examples. (80ms MFCC features)

The same experiment can be made with the 4000 examples coming from all four speakers, to demonstrate that the method is, in some way, robust to multi-speakers learning. In this experiment, the training sessions are 200 examples long and after each training session the process is tested with a constant set of 400 examples: 100 from each speaker. The training set is a succession of 900 examples from each speaker, presented by order of speakers. Such results are presented in figure 3.

These experiments show the good accuracy of our system on the keyword recognition problem. We may compare these results with those from Bosch et al. (ten Bosch et al., 2008) within the ACORNS project, whose database we used. Actually our results are quite similar to the ones they obtained us-

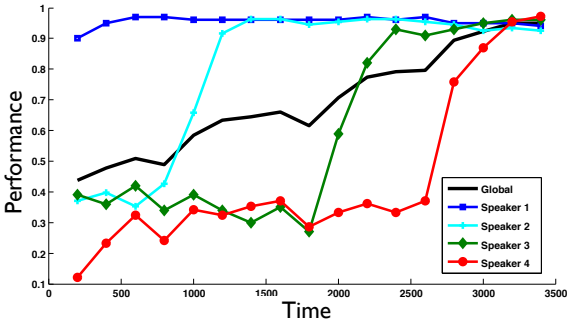

Figure 3: Success rate against number of training examples for incremental learning and testing, 4000 examples: 3800 for training and 200 for testing, one 200 examples test cycle each 100 train examples. (80ms MFCC features) One curve represents global accuracy; the other ones are for each speaker, accuracy reached on the test examples from this speaker.

ing non-negative matrix factorization, which method is also not centered on segmentation and proved to reach maximal performances among a variety of various competing technical approaches.

Those results demonstrate, first of all, the ability of our system to build an internal representation of speech units, in an unsupervised manner (information about keywords is not used in the building of the dictionary), and then to use this internal representation to achieve a keyword recognition task, performed by a kind of semantic engine, which in our experiments is the score system.

4.3 Limitations and further directions

We have presented in this article a framework for autonomous discovery of speech invariants that are useable for speech recognition. This framework essentially builds a new representation of the input signal, as a bag of discretized acoustic features. However despite the word *bag* inherited from text processing, it is completely possible to keep position information on the feature. In this article we have ignored this information in order to demonstrate the efficiency of the extracted local features but it would be of interest to take into account sequential information in future work. For example, hidden Markov models may be built on top of the introduced discretized acoustic features, or sequence analysis methods.

Furthermore, our framework as it is presented is a pre-processing, bringing a new sound representation which is useable with a wide variety of existing methods with a significant complexity reduction from the original input signal.

It is also important to notice that the current clustering method is not completely optimal. Actually the structure of the DTW distance is not completely exploited and experimental analysis shows

that the built clustering has difficulties to cover the whole MFCC sequence space. This could be improved by using better cluster representation, for example using adapted kernels for DTW, such as in (Shimodaira et al., 2001).

The presented framework offers the ability to separately improve each one of these components. For example, the clustering method we used can be replaced by, and thus easily compared to, non-negative factorization.

Finally, we might mention that this framework yet only targets recognition. It would thus be an important further development to integrate a generative model in order to combine perception and action.

References

Aimetti, G. (2009). Modelling Early Language Acquisition Skills : Towards a General Statistical Learning Mechanism. In *EACL (Student Research Workshop)*, pages 1–9. The Association for Computer Linguistics.

Brandl, H., Joublin, F., and Goerick, C. (2008). Towards unsupervised online word clustering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5073–5076, Las Vegas, NV.

Brent, M. R. (1999). Speech segmentation and word discovery: A computational perspective.

Ellis, D. P. W. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab.

Farrell, K., Mammone, R. J., and Gorin, A. L. (1993). Adaptive language acquisition using incremental learning. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1:501–504.

Filliat, D. (2008). Interactive learning of visual topological navigation. In *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*.

Gajjar, M. R., Govindarajan, R., and Sreenivas, T. V. (2008). Online Unsupervised Pattern Discovery in Speech using Parallelization. In *InterSpeech*, pages 2458–2461. ISCA.

Gorin, A. L., Levinson, S. E., and Sankar, A. (1994). An experiment in spoken language acquisition. *Speech and Audio Processing, IEEE Transactions on*, 2(1):224–240.

Gorin, A. L., Petrovska-Delacretaz, D., Wright, J., and Riccardi, G. (1999). Learning spoken language without transcription.

Iwahashi, N. (2003). Language acquisition through a human-robot interface by combining speech, visual, and behavioral information. *Inf. Sci. Inf. Comput. Sci.*, 156(1-2):109–121.

Joachims, T. (1997). Text Categorization with Support Vector Machines: Learning with Many Relevant Features.

Kuhl, P. K. (2004). Early language acquisition: cracking the speech code. *Nature Reviews. Neuroscience*, 5(11):831–43.

Ma, B. and Li, H. (2005). Spoken language identification using Bags-of-Sounds. In Dong Minghui, Li, H., and Zhang, M., (Eds.), *International Conference on Chinese Computing*, Singapore.

Nister, D. and Stewenius, H. (2006). Scalable Recognition with a Vocabulary Tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA. IEEE Computer Society.

Park, A. S. and Glass, J. R. (2008). Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech and Language Processing*, 16(1):186–197.

Roy, D. and Pentland, A. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science*, 26(1):113–146.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Shimodaira, H., Noma, K.-i., Nakai, M., and Sagayama, S. (2001). Dynamic time-alignment kernel in support vector machine. In *Neural Information Processing Systems*, volume 2, pages 921–928.

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. *International Conference on Computer Vision*, page 1470.

ten Bosch, L., Boves, L., Hamme, H. V., and Moore, R. K. (2009). A Computational Model of Language Acquisition: the Emergence of Words. *Fundam. Inform.*, 90(3):229–249.

ten Bosch, L. F. M., van Hamme, H., and Boves, L. W. J. (2008). Unsupervised detection of words questioning the relevance of segmentation. In *Speech Analysis and Processing for Knowledge Discovery*, ITRW ISCA. Bonn, Germany.