Accélérer et simplifier la reconnaissance d'objets avec des descripteurs visuels et contextuels simples

Guido Manfredi, Michel Devy, Daniel Sidobre

Accélérer et simplifier la reconnaissance d'objets avec des descripteurs visuels et contextuels simples

G. Manfredi[1] M. Devy[1] D. Sidobre[1,2]

[1] CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
[2] Univ. de Toulouse, UPS, LAAS
mail : gmanfred, michel, daniel@laas.fr

Résumé

Cet article se propose de mettre en évidence que les techniques de reconnaissance d'objets classiques sont trop complexes pour affronter des cas réels en Robotique, typiquement pour reconnaître et localiser la miriade d'objets présents dans un milieu humain. Nous montrons que des descripteurs de données simples, exploitant des données visuelles, permettent d'accélérer la reconnaissance (classes ou instances) au sein de grandes bases d'objets appris au préalable. Dans une base d'objets candidats nous montrons que jusqu'à 90% des candidats peuvent être écartés rapidement et sans erreur par des descripteurs visuels simples. Le choix final peut alors être fait à moindre coût avec des méthodes complexes. Quand ces fonctions sont intégrées sur un robot d'assistance à l'Homme dans son domicile, les cas les plus difficiles peuvent être grandement simplifiés en exploitant des informations contextuelles acquises lors de l'exécution d'une tâche par le robot.

Mots Clef

Reconnaissance d'objets génériques, descripteurs globaux, classification hiérarchique, données RGBD, couleur, contexte

Abstract

This paper puts forward the fact that current object recognition methods are too complex for real robotics applications, for example to recognise the thousands of everyday objects used by humans. We show that simple descriptors based on visual cues make the recognition problem tractable for huge known objects databases. Among a database of 300 candidat objects, up to 90% of the candidats can be discarded quickly with little error using simple visual cues. The final choice among the remaining candidats can then be done with more complex methods. When these functions are integreted to a robot, the hardest situations can be greatly simplified using contextual information acquired online.

Keywords

Generic object recognition, global descriptors, hierarchical classification, RGBD data, color, context

1 Introduction

De plus en plus de tâches industrielles sont réalisées par des robots mais peu à peu on voit aussi apparaître des robots dans les foyers pour accomplir des tâches domestiques ; le défi est d'étendre les capacités d'autonomie de ces robots pour assister des personnes handicapées ou âgées. Afin d'être autonome, tout en étant accepté par le public, un robot doit pouvoir exécuter des tâches sans aide extérieure : sans modification des lieux ou sans opérateur. Cette autonomie passe d'abord par la compréhension du monde qui l'entoure ; ce monde est constitué de petits objets manipulables par l'Homme ou le Robot (couverts, verres, bibelots…), de grands objets déplaçables (tables, chaises, portes…) et d'objets fixes (meubles, murs, sol…) que le robot doit être capable de segmenter et reconnaître. Après avoir repéré les surfaces planes sur les objets fixes ou sur les tables, le robot pourra manipuler les objets posés dessus. On s'intéresse dans la suite aux techniques de reconnaissance et localisation de ces objets petits manipulables.

Les méthodes classiques de vision par ordinateur utilisent l'apparence (couleur et texture) ou la forme de l'objet pour l'identifier de manière unique depuis une simple image. Récemment, l'apparition de caméras RGBD (Kinect), de caméras 3D (optique à temps de vol) à faible coût a également favorisé l'utilisation de techniques de reconnaissance fondées sur des nuages de points 3D. Toutes ces méthodes sont très gourmandes en temps de calcul et puissance informatique. En situation réelle, le robot est amené à voir plusieurs dizaines d'objets à chaque instant et doit être capable de reconnaitre plusieurs centaines d'objets differents. Comme nous allons le voir par la suite, les approches classiques de reconnaissance d'objet utilisent des models trop complexes pour faire le tri rapidement au sein de grandes bases de données.

Par ailleurs, un robot possède d'autres sources d'information que le nuage de points correspondant à l'objet. Il possède des connaissances contextuelles : il identifie le lieu - cuisine, salon, chambre…- dans lequel il évolue, les objets qu'il observe ne sont pas disposés au hasard mais selon l'utilisation qu'en font les humains. Il se peut même que le robot observe l'objet au cours de cette utilisation, donc reconnaisse les gestes de l'Homme avec l'objet. Le robot

FIGURE 1 – Sans le contexte, les objets sont difficilement identifiables (colonne de gauche). L'information de contexte peut venir du lieu (haut), d'un objet couramment associé (milieu) ou de l'utilisation qu'en fait l'humain (bas).

connait ou peut apprendre le lien entre le contexte et la nature des objets (Figure 1) ; l'identification du contexte lui fournit donc une grande quantité d'information permettant une reconnaissance, plus facile et plus robuste, des objets perçus. Cet article propose un pré-traitement simple qui, aidé du contexte, permettent d'assurer une reconnaissance rapide sans sacrifier à la robustesse.
Ce document est organisé comme suit : nous verrons d'abord l'état de l'art en matière de reconnaissance visuelle d'objets et la structure de l'approche classique. Puis, nous verrons les différentes implémentations disponibles permettant la reconnaissance d'objets. Par la suite, nous proposerons une méthode basée sur une cascade de descripteurs visuels et contextuels simples pour limiter le nombre de candidats dans la base d'objets connus pour un objet observé. Nous décrirons les résultats préliminaires des expériences menées pour valider notre approche, afin de montrer les avantages de notre méthode par rapport aux approches classiques. Pour finir, nous conclurons et présenterons nos futurs travaux, avec les améliorations et extensions de la méthode que nous développons.

2 Travaux antérieurs

D'après [15], la reconnaissance de classes (cet objet est une tasse), d'instances (cet objet est ma tasse) et de pose (cette tasse est tournée dans telle direction) sont différentes applications d'un même problème plus général que nous appellerons par la suite reconnaissance d'objets.
Afin de reconnaître un objet, il faut d'abord créer un modèle perceptuel de cet objet ; le modèle CAO n'est généralement pas adapté ou suffisant, puisque l'apparence n'y ait pas représentée. Il faut d'abord acquérir un grand nombre d'images sur l'objet isolé, en parcourant au mieux tous les points de vue potentiels. Puis, pour chacune de ces images, il existe deux façons pour caractériser l'objet depuis ce point de vue : apprendre un descripteur global qui caractérise l'objet entier, ou apprendre de nombreux descripteurs locaux qui caractérisent les parties remarquables de l'objet. On parle de descripteurs locaux ou globaux (Figure 2). Or l'objet tel qu'il sera vu dans une scène, pourra avoir une attitude ou une apparence différente de celles enregistrées dans son modèle, e.g. il pourra être partiellement occulté. C'est pourquoi les descripteurs utilisés pour créer le modèle doivent posséder des propriétés d'invariance qui les rendent résistants aux changements d'illumination, à l'occultation, au bruit, au flou, aux changements d'échelle ou encore aux rotations, voire même aux transformations affines [18].

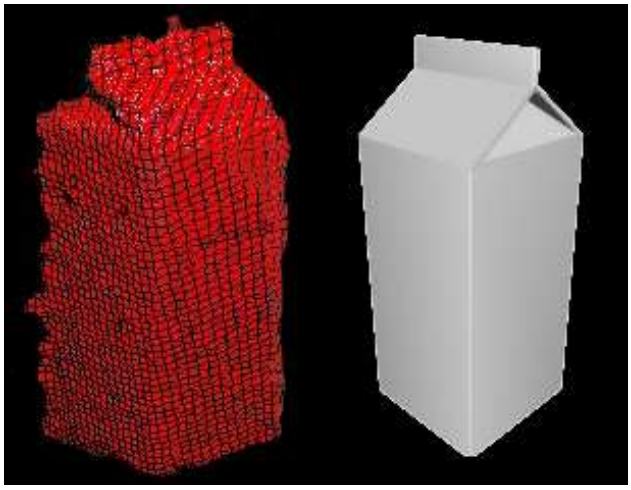

FIGURE 2 – Un modèle constitué d'un nuage de descripteurs locaux (gauche) et un modèle fait d'un descripteur global (droite).

Pour reconnaître un objet depuis une image acquise sur une scène, une première étape consiste à extraire des régions de l'image dans lesquelles l'objet pourrait se trouver. Puis les descripteurs choisis sont extraits de ces régions et comparés, à l'aide d'une distance prédéfinie, avec les descripteurs du modèle de cet objet. S'il existe de nombreux appariements entre descripteurs extraits de l'image et appris du modèle, alors le système en déduit que l'objet correspondant est présent dans la scène perçue ; une pose approximative peut aussi être déduite de ces mises en correspondance.

Historiquement, les premières approches ont exploité les contours extraits de l'image de l'objet (internes ou silhouettes) pour créer un modèle ; une mesure géométrique entre descripteurs de contours a été proposée pour éviter les ambiguïtés [9]. Mais ce type d'approches rend la détection sensible aux changements de luminosité, au bruit, au flou et aux occultations. Pour gagner en invariance à la

Ref.	Type descripteurs	Organisation
[13]	Global	Hiérarchie intra-classe de patrons
[21]	local	Hiérarchie inter-classes
[15]	Global	Hiérarchie classe-instance-pose

TABLE 1 – Type et organisation des descripteurs.

luminosité, [13] propose d'utiliser une transformée, la Distance Transform (DT). Dans [16], l'auteur utilise un détecteur de points remarquables à base de différences de gaussiennes (DoG) et une caractérisation de ces points par des descripteurs dits Scale Invariant Feature Transform (SIFT), qui codent les orientations du gradient dans un voisinage. Ce codage a été généralisé dans le descripteur HOG [12] afin de caractériser une région. Ces descripteurs HOG sont couramment exploités pour la détection de personnes ou de piétons de manière générique, mais aussi pour reconnaître des objets génériques tels que des voitures (PASCAL VOC Challenge). Pour ces problèmes d'indexation d'une base d'images en fonction de la présence d'un objet particulier, ils peuvent être associés aux descripteurs de régions fondés sur les points d'intérêt, tels que les BoW (Bags Of Words) [11]. Toutes ces méthodes fondées sur l'apparence, ont l'avantage d'offrir de fortes invariances à la luminosité, mais aussi dans une certaine mesure, aux rotations et changements d'échelle [18]. De plus, comme elles se fondent sur des descripteurs locaux, elles offrent une résistance partielle aux occultations.

Enfin, avec l'introduction de capteurs 3D, les descripteurs utilisant l'information 3D se sont répandus [7]. Dans [22], les auteurs utilisent un principe similaire à celui qui a permis de créer les descripteurs 2D SIFT, mais le généralisent au cas 3D. Par ailleurs, dans [20], les auteurs proposent un descripteur dépendant du point de vue. Ces dernières générations de descripteurs sont robustes mais le temps nécessaire à les calculer les rend difficilement utilisables dans une application robotique temps réel [7] [15].

Par ailleurs, toutes ces approches souffrent d'une augmentation de la complexité avec le nombre d'objets à détecter, en particulier dans le cas où le robot cherche à identifier tous les objets présents dans une scène. En effet, il faut comparer les descripteurs extraits avec tous les modèles possibles [13] [21] [15]. Une première approche dans [13] a été d'organiser les descripteurs d'une classe donnée dans un arbre pour réduire la combinatoire de la recherche. L'auteur de [21] propose de réunir les modèles qui partagent des descripteurs communs pour factoriser la recherche. Dans [15], les auteurs proposent de former une structure d'arbre à partir des descripteurs pour résoudre simultanément les problèmes de reconnaissance de classes, d'instances et de poses.

Ces méthodes, regroupées dans le Tableau 1, permettent d'accélérer les traitements mais, avec les descripteurs actuels, elles impliquent des traitement lourds en termes de temps de calcul, surtout pour des bases de données de grande taille. Rappelons que, dans le contexte de la robotique d'assistance, le robot doit être capable de reconnaître, à une vitesse acceptable pour l'humain, tout objet présent dans son domicile. Ceci implique qu'il dispose d'une base de données pouvant atteindre plusieurs centaines, voire milliers, d'objets usuels ; base qu'il doit être capable d'étendre s'il découvre de nouveaux objets.

3 Evaluation de méthodes disponibles.

Nous traitons de la reconnaissance d'objets par Vision dans un contexte Robotique. Notre objectif à terme sera donc d'intégrer cette fonctionnalité visuelle dans un système robotique complet, et de valider nos travaux par des expérimentations sur des tâches qu'un robot effectuera en exploitant les résultats de la reconnaissance et de la localisation d'objets, typiquement la saisie d'objets par un manipulateur, ou le positionnement d'un robot vis-à-vis d'un objet. Pour ce faire, nous avons évalué des algorithmes de perception existants vis-à-vis de leur intégration sur un robot. Les critères pour qu'un logiciel soit utilisable dans un contexte robotique est qu'il fonctionne à une vitesse inférieure à la seconde et que la création de nouveaux modèles soit possible pour qu'un robot puisse apprendre de nouveaux objets. Nous avons testé et tenté d'intégrer cinq logiciels Open Source.

La première approche est un système de reconnaissance et localisation d'objets plans utilisant des fonctions de la bibliothèque OpenCV [4]. Il se base sur un détecteur FAST [19] et des descripteurs BRIEF [10]. La recherche de points appariés par la méthode RANSAC (les appariements cohérents votent pour une même homographie) permet d'obtenir avec précision la position d'un objet plan. Bien que limité à des objets plans, ce programme s'exécute en moins de 100ms et permet d'intégrer facilement de nouveaux modèles d'objets. Le programme peut être trouvé ici [2]. La complexité de la reconnaissance est celle de l'étape de mise en correspondance, faite par plus proche voisin, qui est linéaire $O(Nd)$, où N est le nombre de points d'interets présents dans la base de données (proportionel au nombred 'objets) et d est le nombre de dimensions des descripteurs utilisés.

La deuxième approche évaluée est MOPED (Object Recognition and Pose Estimation for Manipulation) [17]. Elle propose une approche basée sur les points d'intérêt 2D SIFT pour une détection robuste et rapide. Ces points 2D sont regroupés spatialement par une méthode des K-means afin de robustifier la reconnaissance. MOPED est mis à disposition par Carnegie Melon University sur ce site [3]. Néanmoins, la rapidité de l'implémentation mise à disposition, vient surtout de l'utilisation de processeurs GPU. Or, comme la bibliothèque GPU exploitée dans cette version est aujourd'hui obsolète, cela rend cette implémentation

bien trop lente. Par ailleurs, la création de nouveaux modèles d'objets pour ce logiciel est loin d'être aisée. Cela passe par une utilisation de la bibliothèque Bundler [1], qui regroupe des fonctions nécessaires pour reconstruire un modèle 3D à partir des mouvements d'une caméra. Notons que un de nos travaux actuels, est l'intégration de Bundler dans une fonction de modélisation et reconnaissance fondée sur des descripteurs locaux, fonction complexe qui serait exploitée uniquement en cas d'échec de la méthode mise en avant dans cet article, uniquement fondée sur des descripteurs globaux. Là encore, la complexité de la reconnaissance est équivalente à celle de l'étape de mise en correspondance, faite par plus proche voisin, qui est linéaire $O(Nd)$.

La troisième approche qui a été évaluée est Linemod [14], qui est un logiciel intégré à OpenCV par l'Ecole Polytechnique Fédérale de Lausanne. Il utilise des descripteurs de la famille des HOG avec des techniques de template matching pour détecter les objets à partir de leurs silhouettes fermées. Aucun modèle n'est fourni avec ce programme, mais un programme de modélisation est disponible pour en rajouter. Les modèles appris sont sensibles aux caractéristiques de la caméra utilisée et aux conditions d'illumination. Ceci rend l'approche peu pratique pour une utilisation par un robot qui se déplace dans des environnements soumis à de fortes variations d'illumination, et qui peut utiliser plusieurs types de caméras pour reconnaître des objets. Par ailleurs, cette technique se décline en deux approches : avec ou sans information de profondeur. Sans information de profondeur la méthode n'est pas suffisamment robuste pour notre application. Avec information de profondeur, elle nécessite une camera 3D (Kinect, Xtion, etc.) pour atteindre une robustesse acceptable. Pour cette approche, la complexité de l'étape de reconnaissance n'est pas claire. Néanmoins, d'après les auteurs, un objet est typiquement représenté par 2000 patrons. Et il faut 100ms pour reconnaitre le bon patrons parmis 2000. D'après les graphiques présentés, l'évolution de la complexité semble linéaire en fonction du nombre de patrons.

Parmi les méthodes mises à disposition avec le middleware Robot Operating System (ROS) [6] a été proposé un noeud de reconnaissance d'objet à l'aide d'un capteur de profondeur. C'est la quatrième solution que nous avons évaluée. Elle s'applique uniquement à la reconnaissance d'objets symétriques de révolution posés debout sur une surface plane, typiquement des bouteilles, verres, tasses, canettes...A partir d'un nuage de points fourni par une camera 3D, une surface plane est segmentée avec une approche de type RANSAC. Les points restants (ceux qui ne sont pas dans le plan majoritaire) sont regroupés en régions, avec une méthode de clustering fondée sur la distance euclidienne. Enfin chaque région est comparée par une méthode ICP avec les modèles 3D d'objets présents dans une base de données. Si un objet n'est pas reconnu, une boîte englobante est calculée pour le délimiter grossièrement. Malheureusement, les façons d'ajouter des objets supplémentaires ou de savoir quels sont les objets présents dans la base sont obscures ce qui rend cette méthode difficilement utilisable. De plus, il faut plusieurs secondes pour réaliser toutes les étapes aboutissant à la reconnaissance de l'objet ; ceci limite l'intérêt de la méthode dans un contexte robotique. Cette technique aussi utilise une étape de plus proche voisins entre les points 3D des models connus et les points visibles. Encore une fois, la complexité evolue de manière linéaire.

Ref.	Dimension /descripteur (d)	Descripteurs /objet (N)	Complexité
[10]	64	5000	O(Nd)
[17]	256	3000	O(Nd)
[14]	>126	2000	O(Nd)
[6]	3	2000	O(Nd)
[22]	352	6000	O(Nd)

TABLE 2 – Les méthodes de l'état de l'art utilisent des descripteurs de haute dimensionalité en grand nombre.

Une dernière option que nous avons testée, passe par l'utilisation de la bibliothèque de traitement de nuages de points Point Cloud Library (PCL) [5]. Celle-ci met à disposition de nombreux descripteurs de points 3D ainsi que des méthodes de recalage entre nuages de points 3D (typiquement un nuage extrait et un nuage modèle 3D). Ces techniques se sont avérées les plus robustes mais au prix d'un temps de calcul de plusieurs secondes pour la moindre reconnaissance d'objet. Cette dernière méthode utilise aussi la technique des plus proches voisins et à donc au moins une complexité linéaire.

Toutes les techniques décrites ont une complexité linéaire. Bien que la methode des plus proches voisins puisse être simplifiée pour avoir une complexité inférieur, dans les pire des cas la complexité est toujours linéaire. Or, quand on doit utiliser les plus proches voisins avec des descripteurs de haute dimensionalité dans des très grandes bases de données, on se ramène au pire des cas. Ces informations sont recapitulées Table 2.

Devant la complexité de toutes ces techniques, leurs temps de traitement souvent trop longs et l'utilisation de modèles trop complexe, nous proposons une approche 3D pour réduire le nombre de candidats dans la base d'objets de manière rapide et robuste.

4 Méthode

Pour simplifier la recherche, nous proposons d'utiliser des descripteurs simples permettant de discriminer les différents objets à reconnaître. L'utilisation du contexte dans lequel l'objet est perçu, permet de tolérer des descripteurs d'une plus grande simplicité pour l'objet lui-même.

Nous implémentons, à l'aide de la bibliothèque PCL, une segmentation d'objets à partir d'un nuage de points. Tout d'abord, les plan présents dans le nuage sont segmentés à l'aide d'une méthode RANSAC. Puis, les points restants

FIGURE 3 – Le résultat après segmentation des plans, segmentation des objets et calcul de la boîte englobante.

sont regroupés en fonction de leur distance euclidienne les uns des autres. Chaque groupement est un objet potentiel segmenté. Par la suite, on ne s'interesse plus à la segmentation et on suppose que l'objet est correctement segmenté. Pour réduire à la fois le coût de reconnaissance et d'apprentissage, nous exploitons une bases de données dans laquelle les vues des objets sont décrites par des descripteurs globaux, définis uniquement par des boîtes englobantes. Chaque vue d'un objet est donc décrite par deux descripteurs : (1) un descripteur global à trois dimensions, taille de la boîte englobante de volume minimum dans l'espace 3D (Figure 3). Celle-ci est calculée en utilisant la technique décrite dans [8]. (2) Par ailleurs, et afin de profiter de l'information couleur que fournit la caméra RGB, nous définissons un descripteur couleur similaire : chaque vue d'un objet est aussi caractérisée par la boîte englobante dans l'espace HSV. Dans les deux cas, pour être robuste aux changements de points de vue, on utilise en plus des dimensions du cube englobant, les variances, selon les trois dimensions du cube, en fonction du point de vue.

Les descripteurs sont organisés de manière hiérarchique pour traiter la reconnaissance par une cascade de classifieurs. Une région segmentée d'une image voit d'abord les dimensions de sa boite englobante comparées avec celles des objets présents dans la base d'objets. Nous utilisons une décision binaire, l'objet est soit un candidat potentiel, soit il est éliminé de la liste. Ce premier tri fait à l'aide d'un descripteur simple, et donc rapide, permet de supprimer un grand nombre d'hypothèses. La distance entre descripteurs prend en compte l'incertitude ou la variabilité intra-classe, évaluée au moment de l'apprentissage ; un niveau de confiance ajustable permet à la méthode d'être plus ou moins sélective, un niveau de confiance plus élevé donnera moins de chance d'éliminer le bon objet à un des étage de la cascade, mais laisser passer plus de faux positifs. Puis, la région est similairement discriminée en fonction de sa couleur. A ce stade, un grand nombre d'objets de la base, candidats pour correspondre à celui perçu dans cette région, ont été écartés : cette région ne peut correspondre qu'à un

FIGURE 4 – Une partie des objets présents dans la base de données

sous ensemble de faible taille de vues apprises. Le context est ensuite utilisé pour supprimer les candidats qui ne correspondent pas au contexte présent : pour un lieu donné, seul les candidats qui peuvent apparaitre dans ce lieu seront conservés.

On peut alors terminer en utilisant n'importe laquelle des méthodes de l'état de l'art décrite précédement. Le nombre de candidats restant étant faible, la complexité de la méthode n'est plus un problème. Le principal but de cet article étant de montrer le nombre de candidats pouvant être éliminés à moindre coût, cette dernière étape de reconnaissance complexe n'est pas inclue lors des experiences.

5 Expériences

Nous illustrons la méthode décrite dans cet article par des résultats expérimentaux. On se propose de montrer que l'exploitation conjointe d'indices visuels simples et d'informations contextuelles suffit pour filtrer un grand nombre d'hypothèses, ce qui permet de limiter les méthodes complexes de reconnaissance sur des descripteurs locaux, à un faible nombre d'instances ou classes d'objet. Nos utilisons la base de données construite dans [15]. Celle-ci est constitué de plus de 100.000 nuages de points pris depuis différents points de vue de 300 objets. Ces objets sont répartis en 51 classes. Les nuages de points sont obtenus par une camera Kinect. Il est utile de noter que les classes sont organisées de manière sémantique et non par rapport à une mesure objective. Par exemple un ballon de rugby est dans la même classe qu'un ballon de foot. On verra plus tard que cette organisation pose problème lors de la classification.

Le calcul des boîtes englobantes est fait à l'aide de l'algorithme présenté dans [8], qui calcule des MVBBs, pour *Minimum Volume Bounding Box*. La force de cette approche réside dans sa complexité algorithme $O(nlogn+n/\epsilon^3)$ dépendante de l'erreur ϵ faite pour le calcul de la boîte. Cette erreur est réglable pour accélérer ou rendre plus précis le calcul. Cet algorithme peut échouer pour certaines configurations remarquables. Lorsqu'une des ces configurations est rencontrée dans la base de données, le nuage de points est supprimé de la base. Par ailleurs, le calcul aboutit parfois à des boîtes englobantes dont une dimension est nulle.

Ces exemples là sont également écartés.
Pour le context, on utilise le lieux où se trouve l'objet. On associe manuellement chaque objet de la base de données à un nombre de lieux où il peut être trouvé, en fonction du bon sens.
La dimension de la base de données nous permet d'évaluer notre proposition à grande échelle, de manière significative, à travers trois expériences. Dans la première, on s'intéresse à l'idée intuitive que les dimensions des boîtes englobantes sont fortement liées aux points de vue. Les boîtes englobantes sont calculées pour chaque pose de chaque instance. On calcule la moyenne et la matrice de covariance des dimensions des boîtes englobantes calculées pour chaque instance. Par souci de clarté, nous présentons les trois instances à plus grande et faible écart type. Nous présentons des résultats pour les boîtes englobantes spatiales et couleur dans les Tables 3 et 4. Les résultats sont exprimés en mètres. Par example, un écart type de 0.002 signifie que la taille apparente de la boite englobante varie en moyenne de 2 milimètres, selon le point de vue.
La seconde expérience met en évidence la pertinence d'une cascade de classifieurs faibles pour la reconnaissance d'instances ou de classes. Il montre le nombre d'hypothèses écartées grâce à la boîte englobante spatiale et celles écartées par la combinaison des boîtes englobantes spatiale et couleur. Pour la reconnaissance de classes on sépare les instances en une base d'entrainement (75% des données) et une base de test (25% des données). Pour la reconnaissance d'instances, on sépare les poses disponibles avec les mêmes ratios. Pour chaque test de reconnaissance d'une classe ou d'une instance, on calcule la distance de Mahalanobis entre chaque image de la base de test et la boîte (moyenne et matrice de covariance) apprise pour cette classe ou instance. Puis, selon la confiance choisie, on regarde le nombre de candidats restants. On utilise des seuils de confiance de 90%, 95% et 99%. Un seuil de confiance de 99% signifie qu'on a 1% de chances de supprimer le bon candidat lors de cette étape.
Enfin, la troisième expérience évalue la cascade pour une tâche de classification. En fonction des bases d'apprentissage exploitées dans la méthode, nous montrons qu'il est possible de reconnaître des objets génériques. Les résultats sont présentés sous la forme d'une courbe précision-rappel avec pour chaque courbe une classification un contre tous. Par souci de clarté, nous présentons les courbes de six classes représentatives de la base de données.
Notons que faute de temps, le contexte n'a pas pu être pris en compte lors de ces expériences.

6 Résultats

La première expérience montre que la variance est faible pour une instance donnée. Elle est au mieux de quelques millimètres pour des objets à fortes symétries. Elle peut atteindre 10cm pour le *binder*, mais là encore ce n'est que selon une dimension, elle reste faible selon les deux autres. Pour les boîtes couleur, il est difficile de se faire une idée

Instance	std X	std Y	std Z
peach_1	0.00267	0.00164	0.00339
bowl_1	0.00299	0.00226	0.00351
orange_2	0.00263	0.00221	0.00382
shampoo_2	0.05665	0.02096	0.01351
keyboard_5	0.07349	0.02994	0.00467
binder_3	0.06228	0.09661	0.01405

TABLE 3 – Les trois instances avec les écarts types les plus grands, en mètres, quand le point de vue change.

Instance	std X	std Y	std X
orange_3	3.0864	4.26667	4.16493
garlic_2	5.49994	3.35492	2.33083
pear_7	4.07947	3.55293	4.39076
shampoo_2	28.446	24.1258	11.2619
binder_3	34.7907	17.6257	14.5281
dry_battery_6	26.7826	25.3871	19.8527

TABLE 4 – Les trois instances avec les écarts types les plus grands dans l'espace couleur Lab.

de la différence en termes de couleurs. Afin d'aider à la compréhension, des résultats la Figure 5 montre une couleur de base, puis la même couleur avec l'ecart type le plus faible que nous ayons calculé et enfin avec l'écart type le plus fort calculé. La boîte englobante spatiale présente une robustesse au changement de pose, quelque soit l'objet au moins certaines de ses dimensions varient peu. En ce qui concerne la boîte englobante couleur, les performances dépendent fortement de l'instance. Les objets ayant une répartition uniforme des couleurs auront une petite variance. Au contraire, les objets ayant de nombreuses couleurs ou reparties de manière non uniforme sur leur surface (par exemple un *Rubik's cube*), ont une grande variance. Malgré cette limitation, nous verrons que la couleur reste un élément discriminant.
La seconde expérience apporte deux informations. Premièrement, que la couleur permet de discriminer les classes et instances. En effet, elle peut fait passer le nombre de candidats de 131 à 120. Deuxièmement, cette expérience montre que le nombre de candidats peut être fortement réduit par l'utilisation des boîtes englobantes. A tel point que, dans le

FIGURE 5 – Une couleur modifié par l'écart type le plus petit et le plus grand. De gauche à droite : Lab = (75, 75, 75), (72, 71, 71), (48, 50, 55).

Confiance	**90%**		**95%**		**99%**	
Lowest						
Spatial	5	46	5	31	12	31
Spatial+Couleur	5	9	5	18	12	31
Highest						
Spatial	131	136	158	159	205	209
Spatial+Couleur	120	127	147	151	200	203

TABLE 5 – Instances. Première ligne : nb d'instances restantes après le test sur la MVBB spatiale. Deuxième ligne : nb d'instances restantes après le test sur MVBBs spatial et couleur. Pour chaque seuil de confiance, nous donnons le résultat pour les deux instances pour lesquelles il reste le plus et le moins d'hypothèses.

Confience	**90%**		**95%**		**99%**	
Lowest						
Spatial	1	4	2	6	5	12
Spatial+Color	1	4	2	6	5	12
Highest						
Spatial	37	39	41	41	47	47
Spatial+Color	36	38	40	40	46	47

TABLE 6 – Classes. Première ligne : nb de classes restantes après le test sur la MVBB spatiale. Deuxième ligne : nb de classes restantes après le test sur MVBBs spatial et couleur. Pour chaque seuil de confiance, nous donnons le résultat pour les deux classes pour lesquelles il reste le plus et le moins d'hypothèses.

tableau de détection de classes, dans un cas, un seul candidat reste.

La dernière expérience présente les courbes précision-rappel pour six classes en Figure 6. De manière générale, la cascade seule est insuffisante pour la classification. Néanmoins, on note que la classe *keyboard* a de bons très résultats. Ceci peut s'expliquer par le fait que la taille des objets dans cette classe est très différente de celle des autres objets ce qui permet de les discriminer facilement.

7 Conclusions

La contribution principale de ce document est de démontrer par l'expérience que l'utilisation de descripteurs à partir de différentes modalités exploitées en cascade et l'exploitation d'informations contextuelles permettent de traiter la reconnaissance de classes ou d'instances d'objets, avec des techniques faciles à mettre en place et peu coûteuses en termes de complexité algorithmique. En effet, cette approche nous a permis d'obtenir, pour une base composée de 300 objets, une étape préliminaire de reconnaissance robuste.

Nous mettons en évidence que chaque étage de la cascade de descripteurs, permet d'écarter un grand nombre de candidats, ce qui accélère le temps de traitement.

Enfin, pour des situations et des objets plus complexes, l'algorithme décrit dans cet article peut servir de prétraitement à des algorithmes classiques fondés sur la forme 3D ou sur la texture des objets, algorithmes plus précis mais plus coûteux.

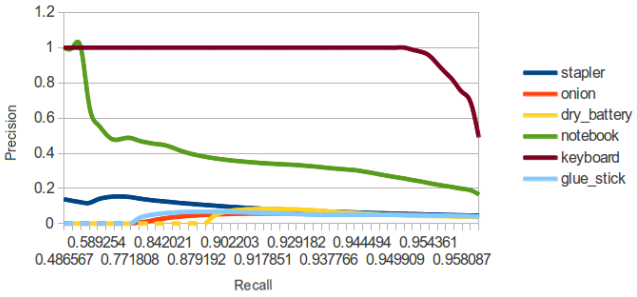

FIGURE 6 – Les coubes de précision-rapel pour six classes représentatives de notre base de données.

8 Travaux futurs

Sur la reconnaissance d'objets, les travaux futurs concernent de nombreux sujets. Tout d'abord, nous traiterons de l'extraction des informations contextuelles comme les gestes accomplis avec un objet, les objets présents autour de celui-ci, le lieu dans lequel le robot est localisé...Pour l'instant, nous annotons les modèles avec les informations contextuelles ; notre objectif est que ces relations objet/objet, objet/lieu ou objet/homme qui forment le contexte, soient apprises par des méthodes d'apprentissage non supervisé. Il s'agira de permettre au robot d'apprendre ces liens par l'expérience ; le robot pourrait ainsi utiliser les liens nouvellement appris pour renforcer ses capacités de reconnaissance.

Nous exploiterons la théorie des croyances pour représenter et mettre à jour l'incertitude (scores de confiance des hypothèses en cours d'évaluation) tout au long du processus de reconnaissance, fondé sur les descripteurs globaux ou locaux et l'information contextuelle.

Les résultats de ces travaux seront exploités pour différentes applications : reconnaître et localiser des objets de la vie courante pour les manipuler (projet ANR ASSIST), guider un robot par la reconnaissance d'objets perçus dans l'environnement (projet régional CAAMVIS) ou encore surveiller un opérateur qui manipule des objets lors d'une tâche collaborative d'assemblage (projet ANR ICARO).

Références

[1] Bundler : Structure from Motion (SfM) for Unordered Image Collections. urlhttp ://phototour.cs.washington.edu/bundler/.

[2] Markerless plan pattern detection. `http://www.packtpub.com/support/10283`.

[3] MOPED : Object Recognition and Pose Estimation for Manipulation. `http:`

//personalrobotics.ri.cmu.edu/projects/moped.php.

[4] Open Computer Vision library. opencv.willowgarage.com/.

[5] Point Cloud Library. urlwww.pointclouds.org.

[6] ROS : Robot Sperating System. urlwww.ros.org.

[7] A. Aldoma, Z. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R.B. Rusu, S. Gedikli, and M. Vincze. Tutorial : Point cloud library : Three-dimensional object recognition and 6 dof pose estimation. *Robotics Automation Magazine, IEEE*, 19(3) :80 –91, sept. 2012.

[8] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38 :91–109, 2001.

[9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4) :509 –522, apr 2002.

[10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief : Binary robust independent elementary features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer Berlin Heidelberg, 2010.

[11] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *Workshop on statistical learning in computer vision, ECCV*, 1 :22, 2004.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 – 893 vol. 1, june 2005.

[13] D.M. Gavrila and V. Philomin. Real-time object detection for ldquo ;smart rdquo ; vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 87 –93 vol.1, 1999.

[14] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2257 –2264, june 2010.

[15] K. Lai, Liefeng Bo, Xiaofeng Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817 –1824, may 2011.

[16] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150 –1157 vol.2, 1999.

[17] M. Martinez, A. Collet, and S.S. Srinivasa. Moped : A scalable and low latency object recognition and pose estimation system. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2043 –2049, may 2010.

[18] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10) :1615 –1630, oct. 2005.

[19] E. Rosten, R. Porter, and T. Drummond. Faster and better : A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1) :105 –119, jan. 2010.

[20] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155 –2162, oct. 2010.

[21] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1481 –1488, june 2011.

[22] Federico Tombari, Samuele Salti, and Luigi Stefano. Unique signatures of histograms for local surface description. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 356–369. Springer Berlin Heidelberg, 2010.