



**HAL**  
open science

# Indexation d'une base de données images : Application à la localisation et la cartographie fondées sur des radio-étiquettes et des amers visuels pour la navigation d'un robot en milieu intérieur

Younès Raoui

## ► To cite this version:

Younès Raoui. Indexation d'une base de données images : Application à la localisation et la cartographie fondées sur des radio-étiquettes et des amers visuels pour la navigation d'un robot en milieu intérieur. Automatique / Robotique. Institut National Polytechnique de Toulouse - INPT, 2011. Français. NNT: . tel-04234303v1

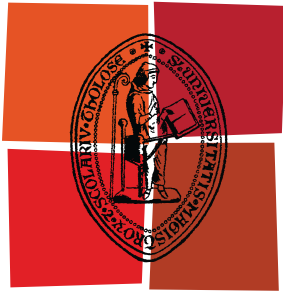
**HAL Id: tel-04234303**

**<https://theses.hal.science/tel-04234303v1>**

Submitted on 20 Oct 2011 (v1), last revised 10 Oct 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université  
de Toulouse

# THÈSE

En vue de l'obtention du  
**DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

**Délivré par :**  
Institut National Polytechnique de Toulouse (INP Toulouse)

**Discipline ou spécialité :**  
Systèmes informatiques

---

**Présentée et soutenue par :**  
Younès RAOUI

**le :** samedi 30 avril 2011

**Titre :**

Indexation d'une base de données images : Application à la localisation  
et la cartographie fondées sur des radio-étiquettes et des amers visuels pour  
la navigation d'un robot en milieu intérieur

---

**Ecole doctorale :**  
Systèmes (EDSYS)

**Unité de recherche :**  
LAAS-CNRS

**Directeur(s) de Thèse :**  
Michel DEVY  
El Houssine BOUYAKHF

**Rapporteurs :**

Fackr Eddine ABABSA ( MdC, Univ. Evry )  
Fackita REGRAGUI ( Professeur des Universités, Univ. Mohamed V Agdal, Rabat )

**Membre(s) du jury :**

Mourad GHARBI ( Président )  
Fackr Eddine ABABSA ( Rapporteur )  
Fackita REGRAGUI ( Rapporteur )  
Ariane HERBULOT ( Membre )



# Table des matières

<b>Résumé</b>	<b>1</b>
<b>Remerciements</b>	<b>3</b>
<b>Dédicaces</b>	<b>4</b>
<b>Glossaire</b>	<b>7</b>
<b>Introduction générale</b>	<b>9</b>
<b>1 Indexation globale des images</b>	<b>21</b>
1.1 Introduction . . . . .	21
1.2 Récupération d'images par leur contenu . . . . .	21
1.2.1 Attribut couleur . . . . .	22
1.2.2 Attribut texture . . . . .	24
1.2.3 Etat de l'art sur les CBIR . . . . .	24
1.3 Développement d'un descripteur global d'image . . . . .	25
1.3.1 Description de l'image par le modèle de couleur Gaussien . . . . .	26
1.3.2 Espaces d'échelle . . . . .	26
1.3.3 Mesure de la couleur: modèle de couleur gaussien de Koendering . . . . .	26
1.3.4 Mesure de la texture . . . . .	28
1.3.5 Mesure conjointe de la couleur et de la texture . . . . .	31
1.4 Descripteur Couleur Texture . . . . .	33

## TABLE DES MATIÈRES

---

1.4.1	Implémentation du descripteur couleur-texture . . . . .	34
1.5	Conclusion . . . . .	39
<b>2</b>	<b>Indexation locale des images</b>	<b>41</b>
2.1	Introduction . . . . .	41
2.2	Etat de l'art . . . . .	42
2.2.1	Indexation par apparence . . . . .	42
2.2.2	Formulation mathématique . . . . .	43
2.2.2.1	Changement d'échelle . . . . .	43
2.2.2.2	Rotation d'images . . . . .	45
2.2.3	Le détecteur de Moravec . . . . .	48
2.2.4	Détecteur de Harris et ses variantes . . . . .	50
2.2.5	Détecteur de régions saillantes . . . . .	54
2.2.6	Filtres orientables (Steerable filters) . . . . .	56
2.2.7	Descripteur SIFT . . . . .	57
2.3	Développement d'un descripteur couleur invariant à l'échelle et à la rotation . .	63
2.3.1	Nature de notre détecteur-descripteur . . . . .	63
2.3.2	Développement mathématique . . . . .	63
2.3.2.1	Calcul d'un vecteur d'échelles caractéristiques . . . . .	64
2.3.2.2	Calcul de l'orientation . . . . .	65
2.3.3	Calcul du descripteur . . . . .	65
2.4	Reconnaissance d'objets . . . . .	66
2.4.1	Généralités . . . . .	66
2.4.2	Résultats expérimentaux . . . . .	70
2.4.3	Evaluation des performances . . . . .	70
2.4.4	Répétabilité . . . . .	70
2.4.5	Discussion : . . . . .	72
2.5	Conclusion . . . . .	74
<b>3</b>	<b>Localisation des robots mobiles à partir de radio-étiquettes</b>	<b>75</b>
3.1	Introduction . . . . .	75
3.2	Localisation . . . . .	77
3.2.1	RFID . . . . .	78
3.2.2	Etat de l'art sur la navigation par RFID . . . . .	81
3.3	Localisation déterministe par RFID . . . . .	82
3.3.1	Localisation par la méthode Range and Bearing . . . . .	83

## TABLE DES MATIÈRES

---

3.3.2	Localisation par des amers visuels ponctuels . . . . .	90
3.3.3	Localisation stochastique par filtrage particulaire . . . . .	90
3.4	Amélioration du recalage par localisation active . . . . .	96
3.4.1	Discussion . . . . .	98
3.5	Conclusion . . . . .	99
<b>4</b>	<b>Localisation et cartographie à partir d’amers visuels</b>	<b>101</b>
4.1	Introduction . . . . .	101
4.2	Introduction à la localisation et cartographie simultanées . . . . .	101
4.3	Localisation par la vision . . . . .	104
4.3.1	Localisation par odométrie visuelle . . . . .	104
4.3.2	Localisation par amers . . . . .	105
4.3.3	Localisation par indexation d’images . . . . .	106
4.4	Principe et formulation mathématique du MonoSLAM . . . . .	107
4.5	MonoSLAM par nos points d’intérêt . . . . .	111
4.5.1	Simulation . . . . .	112
4.6	Navigation par fusion entre radio-étiquettes et points d’intérêt . . . . .	112
4.7	Conclusion . . . . .	115
<b>5</b>	<b>Cartographie des radio-étiquettes</b>	<b>117</b>
5.1	Introduction . . . . .	117
5.2	Modèles géométriques . . . . .	118
5.2.1	Cartes métriques . . . . .	119
5.3	Modèle topologique . . . . .	120
5.3.1	Cartes topologiques . . . . .	121
5.4	Grille d’occupation . . . . .	124
5.5	Cartographie d’un environnement par des position de radio-étiquettes . . . . .	125
5.5.1	Méthode déterministe . . . . .	125
5.5.2	Méthode probabiliste . . . . .	125
5.5.3	Discussion . . . . .	127
5.6	Conclusion . . . . .	129

## TABLE DES MATIÈRES

---

Conclusions générale	131
Perspectives	133
Publications	135
Annexe	137
Bibliographie	147

# Liste des figures

1	Le robot rover qui va naviguer sur Mars; il crée son combustible à partir de l'atmosphère martienne et progresse par déplacements d'un km par jour (à gauche). Bras manipulateur à poste fixe utilisé dans l'industrie (à droite) . . . .	10
2	Robonaut, construit par <i>Dextrous Robotics Laboratory</i> , est un humanoïde conçu pour travailler avec des astronautes (à gauche), Le robot PR2 est une plateforme dédiée à la recherche (à droite) . . . . .	11
3	Contexte du projet COMMROB : développement d'un chariot de supermarché (à gauche), le robot développé par le groupe FZI à Karlsruhe (Allemagne):Interactive Behaviour Operated Trolley (INBOT) (à droite) . . . . .	13
1.1	Système de récupération d'image par le contenu . . . . .	23
1.2	Représentation à différents espaces d'échelle . . . . .	27
1.3	Processus de formation d'une image . . . . .	27
1.4	Les réponses spectrales du modèle gaussien . . . . .	28
1.5	Les réponses spectrales du modèle gaussien . . . . .	29
1.6	Exemple d'un filtre de Gabor à deux dimensions . . . . .	30
1.7	Démonstration d'un filtre de Gabor appliqué aux caractères Chinois de l'OCR.. Quatre orientations sont indiquées sur la droite de 0 °, 45 °, 90 ° et 135 °.L'image du caractère original et la superposition de tous les quatre orientations sont indiquées sur la gauche. . . . .	31
1.8	Processus de calcul de notre descripteur . . . . .	32
1.9	Première figure : Rice Pdf, Seconde figure : l'entrée de l'image de texture . . .	35
1.10	Histogramme (avec 100 bins) des sorties du filtre de Gabor pour 2 fréquences centrales . . . . .	35
1.11	Nombre d'images récupérées en fonction des appariements corrects . . . . .	36



## LISTE DES FIGURES

---

1.12	Indexation d'images sur indices globaux: validation sur images naturelles. . . .	37
1.13	Récupération d'image : de l'image la plus similaire à la moins similaire . . . .	38
2.1	Reconnaissance d'objet par appariement de points d'intérêt entre un modèle et une scène . . . . .	42
2.2	ellipse d'incertitude correspondant à une analyse des valeurs propres de la ma- trice d'autocorrelation . . . . .	50
2.3	Détection de points d'intérêt invariants à l'échelle. Les réponses spectrales du modèle gaussien. En haut, la sélection manuelle de points de Harris multi- échelle correspondant à une structure locale. En bas, les points d'intérêt sélec- tionnée par l'approche de Harris-Laplace . . . . .	52
2.4	Les parties les plus saillantes sont sélectionnées (gauche) en utilisant l'entropie et (droite) en utilisant les pics dans l'entropie versus l'échelle pondérée par la somme des différences absolues au pic(droite) . . . . .	55
2.5	Détection des points les plus saillants et leurs échelles . . . . .	56
2.6	Détection des caractéristique dans l'espace d'échelle en utilisant une pyramide de gaussienne(à gauche) les niveaux voisins d'une pyramide gaussienne sub- octave sont soustraites pour produire des images de différences de gaussienne. .	59
2.7	Une estimée d'orientation dominante peut être calculée en créant un histogramme de toutes les orientations du gradient( pondérées par leurs amplitudes après le seuillage des petites valeurs) et ensuite en trouvant les pics significatifs dans cette distribution ((Low04)) . . . . .	61
2.8	Structure de notre détecteur-descripteur, 2 champs pour la position spatiale, 21 pour l'échelle, 5 pour l'orientation, 9 pour la texture. . . . .	66
2.9	20 images utilisées pour construire lo modèle de Teddy. Il y a 16 images local- isées dans un cercle équatorial. Cette configuration (avec une variation dans le nombre d'images d'entrée) est typique pour nos expériences de modélisation . .	69
2.10	Reconnaissance d'objets appris par notre méthode d'indexation. Les objets sont capturés à partir d'une séquence acquise par une seule caméra ou à partir des bases d'images du <i>Ponce Group Dataset</i> et de l' <i>Amsterdam library of object images</i> . Les objets reconnus sont identifiés par des points rouges et par un cercle bleu. . . . .	71
2.11	Première figure : Répétabilité du Fast Hessian, DoG, Harris-Laplace, Hessian- Laplace en fonction de l'échelle, Seconde Figure : répétabilité de notre détecteur en fonction de l'échelle . . . . .	72

**LISTE DES FIGURES**

---

2.12	La deuxième image a été générée par le changement de la luminance, le contraste et en effectuant une rotation de la première image . . . . .	73
3.1	A gauche, le trolley robotisé. A droite, le robot RACKHAM . . . . .	76
3.2	Schéma du magasin . . . . .	76
3.3	(A gauche) antenne RFID (A droite) tags RFID . . . . .	79
3.4	Champ de réception des 8 antennes montées sur un robot circulaire . . . . .	80
3.5	Un environnement simulé avec les tags en bleu et la trajectoire Xi . . . . .	84
3.6	Prédiction, Observation, Estimation . . . . .	85
3.7	Les positions prédites à partir du modèle du robot ( odometrie ) . . . . .	86
3.8	La variance de $x_{Est}-x_{True}$ pour 10 cycles de robot, L'application des non-observations réduit l'erreur. La ligne rouge est pour les non observation, la ligne bleue est pour les non-observations . . . . .	87
3.9	L'évolution de la distribution des particules( Avec la prise en compte des non-observations, nous gardons uniquement les particules jaunes) . . . . .	87
3.10	Localisation du robot à différentes positions du robot, avec le calcul de la variance de $x, y, \theta$ . . . . .	88
3.11	Recalage de la position du robot par amers visuels ponctuels . . . . .	89
3.12	Simulation stochastique - évolution dans un milieu riche en tags . . . . .	91
3.13	Simulation stochastique - évolution dans un milieu pauvre en tags . . . . .	92
3.14	Simulation stochastique - évolution dans un milieu riche en tags . . . . .	93
3.15	La variation de l'erreur en X et en Y . . . . .	93
3.16	La variation de l'erreur en distance et en $\theta$ . . . . .	94
3.17	Comparaison des performances en variant le nombre d'échantillons . . . . .	94
3.18	Comparaison des performances en variant le nombre de tirages . . . . .	95
3.19	Comparaison des performances en variant le nombre de particules injectées . . . . .	96
3.20	Variance de la position en fonction du nombre des waypoints . . . . .	98
4.1	Modèle graphique du SLAM en ligne, $z_i$ étant la mesure du capteur, $u_t$ l'action, $x_t$ l'état du robot (à gauche). Une carte construite par un capteur de proximité (à droite) . . . . .	102
4.2	Carte (à gauche), Matrice de corrélation (à droite) . . . . .	103
4.3	Tâches effectuées par un robot pour avoir une représentation correcte de son environnement. Le SLAM est une méthode qui fusionne la localisation et la cartographie . . . . .	104

## LISTE DES FIGURES

---

4.4	TBL, Turn Back and Look (à gauche), TBL dans des images panoramiques (à droite) . . . . .	106
4.5	Résultats d'estimation de la position des patches des descripteurs pour une scène intérieure . . . . .	113
4.6	Position prédite de la caméra avec les incertitudes en utilisant l'odométrie uniquement . . . . .	114
4.7	Position estimée du robot en utilisant l'odométrie et les amers visuels . . . . .	114
4.8	Recalage des positions du robot par fusion des RFIDs et des amers visuels . . . . .	116
5.1	Exemple d'une carte géométrique acquise dans une salle de la Cité de l'Espace à Toulouse, à partir d'un télémètre laser plan. . . . .	119
5.2	Carte topologique . . . . .	121
5.3	Grille d'occupation . . . . .	124
5.4	Mouvement du robot, à chaque étape, il réduit la zone estimée du tag . . . . .	126
5.5	Positions estimées des tags représentées avec des étoiles bleues ( Premier algorithme) . . . . .	126
5.6	Un modèle de capteur simplifié de l'antenne du robot . . . . .	127
5.7	Les positions estimées des tags RFID. La couleur du cercle représente la probabilité a posteriori de la position correspondante . . . . .	128
5.8	Erreur sur la position du tag . . . . .	128
5.9	Étapes de chaque pas d'itération de la méthode de Monte Carlo . . . . .	144

# Résumé

Ce mémoire concerne les techniques d'indexation dans des bases d'image, ainsi que les méthodes de localisation en robotique mobile. Il fait le lien entre les travaux en Perception du pôle Robotique et Intelligence Artificielle du LAAS-CNRS, et les recherches sur la fouille de données menées à l'Université de Rabat. Depuis une dizaine d'années, la vision est devenue une source de données sensorielles essentielles sur les robots mobiles: elle fournit en particulier des représentations de l'environnement dans lequel doit se déplacer un robot. Deux types de représentations peuvent être construites. D'une part les modèles géométriques 3D sont constitués de points, de courbes ou de surfaces extraits et reconstruits depuis une séquence d'images; ces modèles permettent au robot de se localiser précisément sous la forme d'une position et d'une orientation dans un repère métrique. D'autre part les modèles fondés sur l'apparence, sont obtenus à partir d'une base d'images acquises lors de déplacements effectués de manière supervisée durant une phase d'apprentissage: des techniques d'analyse de données (ACP, ACI) permettent d'extraire les données les plus pertinentes de cette base; ces modèles permettent au robot de se localiser de manière qualitative ou topologique dans l'environnement.

Concernant la vision, seules les représentations fondées sur l'apparence ont été considérées. Le robot se localise en recherchant dans la base d'images, celle qui ressemble le plus à l'image courante : les techniques exploitées pour ce faire sont des méthodes d'indexation, similaires à celles exploitées en fouille de données sur Internet par exemple. De nombreux travaux en Robotique ont déjà exploité ces techniques: citons au LAAS-CNRS une thèse soutenue en 2004, qui traitait déjà de localisation qualitative dans un milieu naturel à partir d'une base d'images panoramiques, indexée par des histogrammes couleur; plus récemment, des travaux menés à Amsterdam, Rennes, Clermont-Ferrand, Lausanne... traitaient de l'indexation par des techniques d'appariements de points d'intérêt.

Par ailleurs, nous présentons une technique de navigation par RFID (Radio Frequency IDentifier) qui utilise la méthode MonteCarlo, appliquée soit de manière intuitive, soit de manière formelle. Enfin, nous donnons des résultats très préliminaires sur la combinaison d'une perception par capteurs RFID et par capteurs visuels afin d'améliorer la précision de la localisation du robot mobile.

# Remerciements

Cette thèse a été préparée dans le cadre d'une co-tutelle entre l'Université Mohamed V-Agdal, Faculté des Sciences de Rabat, Maroc et l'Institut National Polytechnique de Toulouse, Université de Toulouse, France.

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Informatique, de Mathématiques appliquées, d'Intelligence Artificielle et de Reconnaissance de Formes à la Faculté des Sciences de Rabat et au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS à Toulouse.

Je tiens à remercier du fond du coeur Le Pr BOUYAKHF El Houssine, Professeur d'Enseignement Supérieur à la FSR, mon directeur de thèse de m'avoir accueilli et encadré dans son laboratoire depuis janvier 2005.

Je lui exprime ici ma profonde gratitude pour la qualité de son encadrement, sa patience, ses compétences scientifiques et ses hauts caractères personnels que j'ai beaucoup appréciés tout au long des travaux de ma thèse

Je remercie aussi chaleureusement mon directeur de thèse en France dans le prestigieux laboratoire de Toulouse, le LAAS-CNRS, Monsieur DEVY Michel, Directeur de recherche CNRS, de m'avoir permis de mener mes recherches dans son groupe de recherche (Robotique Action Perception), de m'avoir orienté et encadré avec beaucoup de patience, le long de la préparation de ma thèse. Je lui adresse ici, l'expression de toute ma reconnaissance et de ma profonde gratitude.

Je remercie aussi mes rapporteurs Madame Le Professeur REGRAGUI Fakhita, Professeur d'Enseignement Supérieur à la FSR, et Le Professeur ABABSA Fakhreddine, Maître de conférences HDR à l'Université d'Evry Val d'Essonne, pour le soin avec lequel ils ont lu ce manuscrit, et pour la qualité de leurs critiques.

Mes remerciements vont également au Professeur GHARBI Mourad, Professeur Habilité à la FSR, et au Professeur HERBULOT Ariane, Maître de conférences à l'Université de Toulouse, qui m'ont honoré de faire partie de mon jury de soutenance.

Je remercie mes amis Abdelhak, Imade, Youssef, Wassima, Amine et tous les autres doctorants et chercheurs du LIMIARF et du LAAS pour les très bons moments que nous avons passés ensemble pendant mes années de thèse.

# Dédicaces

*Je dédie ce travail à :*

*Ma grande mère et mes chers parents*

*Mon frère Yasser et ma soeur Soumaya*

*Mes tantes et mes oncles bien aimés ainsi que mes cousins et cousines, avec toute mon affection et mon estime indéfinis*

*A tous mes amis doctorants et collègues en souvenir des bons moments passés ensemble au LIMIARF et au LAAS*

*A tous ceux et celles qui ont bien voulu participer de près ou de loin à la réalisation de cette thèse, qu'ils trouvent tous ici l'expression de mes vifs remerciements et de toute ma reconnaissance.*





# Glossaire

- RFID : Radio Frequency Identifier
- SLAM : Simultaneous Localization and Mapping
- EKF : Extended Kalman Filter
- monoSLAM : Monocular Simultaneous Localization and Mapping



# Introduction générale

Conçu à l'image de l'homme, un robot mobile autonome a des comportements qui imite les plus importantes fonctions de l'Homme, qui sont la perception, la locomotion, l'autonomie décisionnelle et l'interaction intelligente avec l'environnement dans lequel il effectue des tâches. L'accomplissement de ces tâches complexes nécessite l'analyse et la compréhension du fonctionnement d'un être vivant, en particulier de l'Homme. Le processus d'exécution d'une tâche par l'Homme peut être représenté par le schéma simplifié suivant :

Organes de perception → Traitement des informations → Prise de décision →  
Contrôle des articulations → Organes effecteur.

à l'image de l'homme avec son schéma classique

Les 5 sens → Le cerveau → Actions sur l'environnement

En effet, pour le bon accomplissement d'une tâche donnée, qui nécessite l'utilisation d'organes effecteurs, l'homme collecte les informations indispensables décrivant son environnement, à partir desquelles, l'unité de gestion et de contrôle, fournit les commandes (ordres) nécessaires pour obtenir le résultat désiré. De cela nous pouvons juger l'importance de la partie perception, puisque de sa précision et de la qualité des informations qu'elle délivre dépend le fonctionnement d'un robot, en particulier la précision des déplacements pour un robot mobile.

En 1920, le tchèque Karel Capek a introduit le terme « robot », provenant du mot tchèque *robot* qui signifie « travail forcé », afin de désigner à l'origine une machine androïde capable de remplacer l'homme dans certaines tâches matérielles. Les années 70 et 80 du siècle dernier, ont

## Introduction générale

---

connu l'émergence d'un nouveau domaine dans l'automatisme : la robotique. Et depuis, nous assistons à un développement considérable profitant des avancées technologiques des autres domaines tels que la mécanique, l'automatique, la microélectronique, les microprocesseurs, les capteurs et les technologies logicielles.

La figure 1 présente deux exemples de robot conçu par l'Homme, l'un pour l'exploration planétaire, l'autre pour réaliser des tâches en robotique manufacturière. La figure 2 présente des plate-formes conçues pour interagir avec l'Homme dans son propre environnement.



Figure 1: Le robot rover qui va naviguer sur Mars; il crée son combustible à partir de l'atmosphère martienne et progresse par déplacements d'un km par jour (à gauche). Bras manipulateur à poste fixe utilisé dans l'industrie (à droite)

Le robot mobile se caractérise par l'absence de lien mécanique avec un objet de référence (socle pour un manipulateur, câble d'alimentation, fil de guidage...), ce qui ouvre magistralement la porte à de nombreuses applications potentielles. La liberté de mouvements lui confère une autonomie qui lui permet de trouver des utilisations autres que manufacturières.

Ainsi, la navigation est l'une des tâches les plus importantes pour l'autonomie du robot. Elle a pour objectif principal de réaliser un mouvement reliant un point source à un point destination. L'exécution de cette tâche avec un certain degré d'autonomie nécessite l'utilisation d'un ensemble de ressources matérielles (capteurs, moteurs, structure de locomotion, processeurs...), des fonctions logicielles (détection et évitement d'obstacles, suivi de trajectoires, asservissements...) et d'une architecture embarquée assurant une coopération efficace entre composants matériels et logiciels.



Figure 2: Robonaut, construit par *Dextrous Robotics Laboratory*, est un humanoïde conçu pour travailler avec des astronautes (à gauche), Le robot PR2 est une plateforme dédiée à la recherche (à droite)

En fonction de l'application, plusieurs structures de robots sont utilisées. Le choix d'une structure est directement lié à la nature de l'environnement dans lequel le robot doit évoluer, à la nature de la tâche qu'il doit exécuter, ainsi qu'aux performances recherchées qui dépendent à la fois de la qualité des ressources et de leur gestion.

L'autonomie des robots, qu'ils soient mobiles ou à poste fixe, a permis un usage de plus en plus croissant de la robotique qui commence à envahir tous les secteurs de la vie. C'est ainsi, que nous voyons les médecins s'intéresser à la dextérité de la robotique dans des interventions chirurgicales les plus délicates, dans les traitements méticuleux de la radiothérapie; les militaires, aussi, pour le lancement des missiles, des prises de photographies aériennes avec des avions sans pilote. Dans l'aéronautique, on pense déjà à remplacer des astronautes ou les assister lors des sorties extra - véhiculaires dans l'espace. Nous assistons donc à une révolution dans le domaine d'une technologie qui vise à faciliter la vie de l'homme dans ses préoccupations les plus avancées et les plus fines.

Par ailleurs, l'essor considérable que connaît la robotique en matière d'interface de communication entre le robot et l'Homme, ou entre robots, nous permet d'assister à un usage de plus en plus fructueux dans des lieux de vie de l'Homme. Citons, dans le milieu social, le robot d'assistance aux personnes handicapés, ou dans la vie quotidienne, le robot de service dans les supermarchés où l'on vise déjà à aider le client dans ses choix ou à permettre au personnel de gagner du temps. Les aéroports y trouvent aussi une application de choix. En un mot, là

## Introduction générale

---

où l'homme cherche à faciliter sa tâche, et simplifier son rôle, la robotique se présente comme une solution idéale.

Dans notre travail, nous avons participé au projet européen CommRob (pour *Advanced Behaviour and High-Level Multimodal Communication with and among Robots*) qui vise à mettre en place des robots qui interagissent avec les humains dans leur vie quotidienne. Le projet CommRob a commencé en Février 2007; il s'est terminé en Juin 2010.

L'objectif majeur du projet CommRob est de permettre à un robot d'exécuter des tâches de communication avec un utilisateur et aussi de coopérer avec d'autres robots opérant dans le même environnement que l'homme. Dans le contexte applicatif du projet, il s'agit de robotiser un caddy, donc d'assister l'Homme lorsqu'il fait ses courses dans un centre commercial.

Un défi majeur de CommRob est de démontrer que les robots de compagnie peuvent fournir des services de manière efficace et en toute sécurité, ceci dans un environnement complexe et dynamique, partagé par d'autres êtres humains et d'autres robots. En effet, les environnements humains quotidiens peuvent être dynamiques et complexes, en particulier dans des lieux comme les centres commerciaux et les aéroports.

Les objectifs spécifiques du projet Commrob sont rappelés ci-dessous:

- la définition des modes de communication pour les robots. Cette définition est fondée sur des modèles théoriques de la communication humaine et des analyses de cas pertinents de la communication homme-robot. En outre, si les robots agissent selon ces modes de communication issus de la communication humaine, il est plus probable que les humains les comprennent mieux.
- développement de comportements évolués sur un robot qui doit intervenir dans un environnement humain quotidien. Plusieurs classes de comportement seront mises en oeuvre et conduiront à des performances évolutives des robots: la sécurité, les comportements locaux tels que la planification de trajectoire, l'évitement d'obstacles dynamiques et la localisation. Il s'agit en particulier de fusionner les perceptions acquises depuis plusieurs capteurs et de les intégrer dans les sous-systèmes d'interaction et de communication.
- développement de la communication entre l'utilisateur et le robot, donc de la coordination robot-utilisateur grâce à une interface haptique, le suivi visuel, la commande par la parole et l'analyse gestuelle
- construction d'un prototype de robot, de type *caddy* ou chariot. Le chariot prototype sera utilisé dans les supermarchés, les aéroports. . . En plus du transport des marchandises pour un supermarché ou des bagages pour un aéroport, une autre fonction sera

celle d'aider les clients dans leur déplacement, donc le guidage d'un utilisateur dans un environnement complexe, structuré, dynamique (présence d'objets mobiles) et évolutif (les objets statiques peuvent être déplacés). En particulier, le prototype de robot sera capable d'effectuer les dialogues multimodaux avec les humains, par exemple, pour prendre des requêtes, pour informer l'Homme... Afin de partager son environnement avec d'autres robots et humains, le chariot doit également être en mesure d'éviter les obstacles et naviguer en toute sécurité.

Nous avons participé au projet durant 18 mois afin d'étudier la navigation exploitant à la fois des tags RFID répartis dans l'environnement, et des amers visuels, appris sous la forme de points d'intérêt. Du fait de diverses contraintes (la durée trop courte du séjour, les difficultés technologiques...), nous n'avons pas pu intégrer nos travaux sur les démonstrateurs du projet, présentés en figure 3.



Figure 3: Contexte du projet COMMROB : développement d'un chariot de supermarché (à gauche), le robot développé par le groupe FZI à Karlsruhe (Allemagne): Interactive Behaviour Operated Trolley (INBOT) (à droite)

### Approche générale

Dans ce travail, nous étudions la localisation et la cartographie, qui sont des fonctions indispensables en robotique mobile. Chaque robot a besoin de naviguer pour atteindre son but et réaliser une tâche donnée. Pour cela, il doit avoir une carte qui lui permet de représenter son environnement; ce modèle est exploité pour planifier les déplacements vers des buts connus, et pour calculer sa position courante pendant l'exécution de ces déplacements.



## Introduction générale

---

La cartographie doit traiter le problème de l'intégration des informations acquises depuis les capteurs du robot, vers une représentation donnée de l'environnement, appelée **carte**. Cette fonction doit donc donner une réponse à la question suivante: « comment est représenté le monde? ».

La localisation doit traiter le problème de l'estimation de la position du robot relativement à une carte. En d'autres termes, le robot a besoin de répondre à la question: « où suis-je? ».

Ces deux tâches ne peuvent pas être séparées l'une de l'autre. Afin de cartographier l'environnement à partir d'un ensemble d'observations, il a besoin de savoir à partir de quelles positions ces observations ont été faites.

En même temps, il est difficile d'estimer la position d'un véhicule sans connaître sa carte. Le robot pourrait exploiter des capteurs internes (dits proprioceptifs) pour estimer ses déplacements, puis intégrer ces mesures afin de se localiser, mais il est connu que ce type de localisation donne des erreurs cumulatives. Donc pour corriger les erreurs dues aux capteurs internes, le robot a recours à des capteurs extéroceptifs afin de sélectionner des points de repère dans l'environnement. Dans nos travaux, il s'agit d'une caméra pour détecter des points significatifs dans une scène et d'un lecteur RFID équipé d'antennes directionnelles pour détecter des tags éparpillés dans l'environnement. Nous proposons donc deux méthodes de localisation.

◊ L'indexation ou l'extraction de points d'intérêt est une composante essentielle à plusieurs applications de la vision par ordinateur. Si nous utilisons l'approche de vision pour la navigation d'un robot mobile, nous avons besoin de points d'intérêt pour construire une carte, et par conséquent pour ensuite permettre au robot de se localiser. Les points d'intérêt sont des positions spécifiques dans les images, comme les sommets de montagnes, les coins des bâtiments ou des points remarquables dans l'image d'un objet texturé. Ces types de caractéristiques sont souvent connus sous le nom de clés ou de coins; ils sont souvent décrits par l'apparence des pixels pris dans une sous-image (ou patch) centrée autour de la position du point.

◊ Les avancées récentes dans le domaine de l'identification radiofréquence vont permettre dans l'avenir d'équiper chaque objet dans un environnement par des tags RFID. Ces tags sont passifs; ils contiennent des circuits qui obtiennent de la puissance à partir de l'énergie reçue et transmettent leurs identifiants uniques au lecteur. Les RFID font émerger une variété d'applications. Par exemple, l'assistance des personnes handicapées nécessite la reconnaissance de leurs activités quotidiennes. Une des solutions proposées a été de mettre des lecteurs RFID dans leurs vêtements pour identifier les objets touchés. La séquence de ces objets utilisée dans un raisonnement bayésien permet d'estimer les activités de la personne à assister.

### Navigation

Les stratégies de navigation qui permettent à un robot de se déplacer et rejoindre un but sont très diverses. Nous reprenons une typologie établie par Trullies et al, cité dans le cours proposé par D.Filliat de l'ENSTA, Paris: (Fil06). Il existe des stratégies sans modèles et d'autres avec modèles internes:

- **approche d'un objet:** cette capacité de base permet de se diriger vers un objet visible depuis la position courante du robot, puis d'accoster près de cet objet (on parle d'accostage ou *docking* le long d'une table par exemple).
- **Guidage:** cette capacité permet d'atteindre un but qui n'est pas un objet matériel directement visible, mais un point de l'espace caractérisé par la configuration spatiale d'un ensemble d'objets remarquables ou amers qui l'entourent ou qui en sont voisins; ces amers sont visibles depuis la position courante du robot, qui peut donc être guidé par asservissement visuel par exemple;
- **Mouvement associée à un lieu:** cette capacité est la première qui permet une navigation globale, c'est à dire qui permet de rejoindre un but depuis des positions pour lesquelles ces buts ou les amers qui caractérisent son emplacement, sont invisibles; une phase de planification à partir d'un modèle, est indispensable pour décider comment atteindre un tel but;
- **Navigation topologique:** elle exploite les informations dans le modèle interne, qui décrivent les relations entre les différents lieux, et génère une trajectoire sous la forme d'une liste des lieux qui doivent être traversés (par exemple, franchir la porte A, traverser la pièce, franchir la porte B, longer le couloir C... ) ;
- **Navigation métrique:** elle combine les approches locales à un lieu et la navigation topologique, cela pour permettre au robot de planifier des chemins pour atteindre des zones lointaines de son environnement. Pour cela, elle exploite les positions métriques relatives aux différents lieux en plus de la possibilité de passer de l'un à l'autre.

## Introduction générale

---

Notre objectif est de développer des fonctions perceptuelles nécessaires à la robotique de service. En effet, les services que peut rendre le robot à une personne sont :

- La planification et l'exécution de tâches à proximité ou au contact de l'homme (par exemple, pour donner un objet à l'homme);
- L'intégration dans le système robotique, d'une Interface Homme-Machine (IHM) permettant à l'Homme d'indiquer la tâche à exécuter par le Robot et de préciser les modalités de cette tâche, et au Robot de percevoir l'Homme si la tâche nécessite une interaction sensorielle directe (le Robot doit suivre l'Homme, doit lui donner un objet) ou une reconnaissance des comportements de l'Homme pour détecter des situations à risques, pour comprendre ses intentions;
- L'apprentissage de l'environnement partagé entre l'Homme et le Robot.

Nos travaux au LIMIARF à Rabat et dans le groupe RAP du LAAS-CNRS à Toulouse, ont porté sur la perception de l'environnement, en d'autres termes comment le robot peut-il construire des représentations de son environnement de travail? Comment peut-il naviguer dans son environnement? Comment peut-il reconnaître des objets de son environnement?

Dans le cadre du projet COMMROB, l'environnement de travail est un magasin. Les objets que le robot peut trouver sont les produits vendus dans le supermarché. Ces objets peuvent être fixes (affiches, étagères accrochées sur un mur) ou peuvent être déplacés par l'Homme (verres, bouteilles posés sur une table, chaises autour de la table). On peut assimiler le processus de vision du robot à celui de l'être humain. Quand celui-ci rentre pour la première fois dans un magasin, il doit tout apprendre en utilisant ses facultés cognitives. En effet, les mécanismes de l'apprentissage de l'homme sont au coeur de plusieurs études dans les sciences cognitives. L'homme utilise bien évidemment ses yeux et un grand nombre de connaissances qu'il a accumulées depuis son enfance. Puisque la vision humaine est incomparablement plus performante que la vision artificielle., il est impossible de doter le robot de capacités similaires à l'homme. Cependant nous nous sommes inspirés des travaux faits sur la vision et sur la cognition humaine pour la navigation du robot.

D'un autre côté, pour représenter l'environnement du robot, nous avons utilisé les deux approches suivantes :

**Cartographie par amers locaux :** Les amers locaux que nous avons utilisés sont des amers ponctuels et un ensemble de caractéristiques. Pour les amers ponctuels,, notre tâche était facile car il suffit d'utiliser le modèle sténopé d'une caméra (*pin-hole* ou encore, modèle

à projection perspective centrale) pour pouvoir effectuer des tâches comme la localisation de robot. Pour les caractéristiques, nous sommes passés par deux étapes

1. l'extraction des caractéristiques des images prises par une seule caméra, en utilisant notre propre méthode développée initialement pour l'indexation d'images;
2. appariement en utilisant le produit entre les descripteurs.

**Cartographie par les tags RFID** ; Les tags RFID sont mis dans différents endroits de l'environnement. Nous avons utilisé deux méthodes pour déterminer la position des tags.

1. une méthode déterministe;
2. une méthode probabiliste en utilisant le modèle bayésien que nous estimons par un filtrage particulaire.

## Contributions

Dans ce travail, nous avons traité plusieurs aspects de la vision et de la robotique. D'abord, nous avons travaillé sur les descripteurs qui extraient des informations sur le contenu des images. Nous nous sommes inspirés de l'approche MPEG7 considérée comme le standard dans l'indexation multimédia. Nous avons ainsi développé un descripteur décrivant la couleur et la texture dans l'image. Celui-ci utilise le modèle de couleur gaussien proposé par J.Koendering ([GdBSG01](#)).

Ensuite, notre objectif a été de mettre en  $\frac{1}{2}$ uvre un descripteur ( des coins dans l'image ) local qui permet de corriger quelques défauts constatés dans les descripteurs connus de la littérature, comme Harris-Laplace ([MS04](#)) ou la transformée des caractéristiques invariantes à l'échelle (SIFT) ([Low04](#)). Ces derniers n'étaient pas applicables aux images en couleur. Par conséquent, nous avons développé un détecteur et un descripteur de points d'intérêt extraits depuis des images en couleur texturées; ce descripteur doit être invariant à la rotation et à l'échelle. C'est une extension du descripteur de Harris aux images en couleur texturées via l'utilisation du modèle de Koendering. Pour évaluer la robustesse de ce descripteur, nous avons calculé la répétabilité. Une application sur la reconnaissance d'objets a été développée en utilisant la transformée de Hough.

D'autre part, le problème de localisation et cartographie simultanées (SLAM) intéresse de plus en plus les chercheurs de robotique et de vision. Nous avons appliqué les descripteurs précités à ce problème. En d'autres termes, nous avons fait du SLAM monoculaire avec des amers ponctuels, reconstruits depuis nos points d'intérêt. La méthode développée, inspirée

fortement de la littérature (travaux de Davison, Sola, Montiel et autres) permet d'estimer les positions du robot et des amers avec une bonne précision. On commence au début par l'extraction des points de chaque image acquise en monoculaire par une seule caméra; ensuite on effectue le matching entre les points extraits sur deux images successives.

Nous avons traité aussi dans notre thèse la localisation et la cartographie par RFID. En effet, cette approche est justifiée par le coût très modeste des antennes et tags RFID. En outre, étant données leurs petites dimensions, les tags peuvent être distribués en grande quantité, ce qui augmente la précision. Nous avons développé ainsi des méthodes de localisation et de cartographie déterministes et stochastiques qui ont été validées en simulation.

## Organisation de ce mémoire

Les deux premiers chapitres traitent de Vision pour l'Indexation d'images.

Dans le chapitre 1, nous nous intéressons à la recherche automatique de documents visuels (images) dans des bases de données à partir de requêtes relatives au contenu de ces documents. La technique présentée dans ce chapitre, dite d'indexation, se propose d'attacher à une image un descripteur global de son contenu dans le but de mesurer la ressemblance avec les descripteurs correspondants à la requête.

Le chapitre 2 décrit une nouvelle méthode d'indexation d'images. Les points d'intérêt locaux possèdent des propriétés attrayantes comme la distinction, la capacité de compression et d'extraction par un traitement en temps réel. Nous présentons une nouvelle méthode invariante à la rotation et à l'échelle qui permet d'extraire et de décrire des points d'intérêt dans une image en couleur.

Ensuite, nous développons un système de reconnaissance d'objets disposés dans des environnements encombrés, significatifs d'un milieu intérieur dans lequel un robot personnel devra exécuter des tâches.

Les trois derniers chapitres traitent de localisation et cartographie en Robotique Mobile.

Le chapitre 3 présente notre méthode de localisation fondée sur des tags RFID disposés dans un environnement structuré. Nous considérons que le robot est équipé de 8 antennes RFID. Lorsqu'il se déplace, le robot utilise les tags RFID pour recalibrer sa position. Nous proposons dans ce chapitre des méthodes de localisation déterministes et probabilistes qui utilisent des tags RFID dont les positions sont supposées connues. Nous améliorons par ailleurs le recalage avec une technique de localisation active.

Le chapitre 4 aborde le problème de la localisation et de la cartographie simultanées pour un robot équipé d'une seule caméra en utilisant les points d'intérêt que nous avons présentés dans le chapitre 2 en tant qu'amers visuels. Après avoir montré des résultats sur le SLAM visuel avec nos points d'intérêt, nous présentons une simulation sur la prise en compte à la fois des tags RFID et des amers visuels pour que le robot calcule sa position dans un environnement supposé déjà appris.

Enfin le chapitre 5 est consacré à l'étude du problème de la cartographie. La cartographie ou le mapping vise en fait à générer des cartes à partir des mesures de capteur; dans cette branche de la robotique, les algorithmes probabilistes se sont montrés efficaces. Nous présentons les différentes représentations proposées dans la littérature. Puis nous présentons des résultats préliminaires sur la cartographie appliquée pour les radio-étiquettes ou tags RFID: dans les chapitres précédents, leurs positions étaient supposées connues. Ici nous considérons que la position du robot est connue par les amers visuels; ici aussi nous développons aussi deux méthodes, l'une déterministe et l'autre probabiliste et nous montrons les résultats sur un environnement de simulation.

A la fin du manuscrit nous rappelons nos contributions et nous citons quelques perspectives. Nous terminons avec la liste de nos publications.

Dans l'annexe 5.6, nous présentons différentes méthodes de filtrage pour pouvoir estimer un modèle bayésien. Ces outils sont exploités dans le mémoire pour la localisation et le filtrage.



# Indexation globale des images

## 1.1 Introduction

L'image est une représentation visuelle ou mentale de quelque chose, objet, être vivant ou concept. Concrètement, une image en Vision est une représentation spatiale de l'énergie réfléchie par des objets physiques présents dans le champ de vue du capteur (oeil pour l'Homme, caméra pour un Robot).

Des traitements comme la transformée de Fourier permettent de représenter les pixels de l'image selon des modèles périodiques qui sont caractérisés par leurs fréquences et leurs directions. Ces éléments sont souvent exploités en indexation pour représenter l'objet dans des structures de données sous forme d'arbres à titre d'exemple.

Le calcul des descripteurs d'images par apparence, nécessite de déterminer les caractéristiques utiles dans l'image sur lesquelles les futures opérations vont être appliquées.

Nous présentons une nouvelle méthode d'indexation d'image en utilisant des caractéristiques globales qui décrivent la couleur et la texture ([RBD07](#)).

## 1.2 Récupération d'images par leur contenu

La thématique appelée *Content Based Image Retrieval (CBIR)* traite du problème de sélectionner dans une base d'images, celles qui répondent à une requête, par exemple "*Sortir les images d'environnement routier*". Les méthodes de CBIR se fondent sur l'extraction des caractéristiques importantes des images. Ces caractéristiques peuvent être soit locales soit globales. Dans la suite de ce mémoire, nous allons appliquer ces informations pour la localisation et la navigation d'un robot. Mais avant, nous allons décrire notre méthode pour extraire des



## 1. INDEXATION GLOBALE DES IMAGES

---

caractéristiques globales à partir d'images couleur. Les images en couleur texturées sont porteuses d'informations riches et leurs contenus peuvent les décrire entièrement. Nous rappelons qu'avant que le concept de CBIR ne soit mis en oeuvre en 1992 par T. Kato ([Wik](#)), la recherche dans les bases de données images utilisait des propriétés textuelles. Les techniques et les outils de CBIR sont inspirés des domaines de la statistique, de la reconnaissance de formes, du traitement de signal et de la vision par ordinateur. De plus, en CBIR, les mesures statistiques de similarité entre les représentations sont corrélées avec la similarité subjective visuelle.

Le processus de CBIR peut être modélisé par un modèle en 4 couches, comme le montre la figure 1.1. Les étapes sont: l'indexation par le contenu, l'extraction des caractéristiques, la compilation et la recherche par requête. Les invariants peuvent être extraits en utilisant soit les conditions extrinsèques soit les propriétés intrinsèques ce qui est plus pratique.

Pour mesurer la couleur et la texture, nous avons choisi l'approche physique. Dans cette approche, la luminance de l'image est représentée comme une onde électromagnétique. Le processus d'observation est une intégration de la densité de l'énergie de cette onde dans les dimensions spatio-spectrales. Dans la théorie des espaces d'échelles, les sondes ou caractéristiques doivent avoir une forme gaussienne pour éviter les détails extérieurs lorsque l'échelle est plus grossière. On trouve dans la littérature plusieurs espaces d'échelle, tels que: Poisson, Bessel et Relativiste. L'espace d'échelle gaussien est cependant linéaire et lié au processus de diffusion linéaire.

### 1.2.1 Attribut couleur

La luminance ou les valeurs du tri-stimuli peuvent être utilisées directement ou via quelques transformations dans d'autres espaces de couleur pour extraire des informations sur le contenu de l'image.

La couleur peut être définie comme le résultat perceptuel de la lumière lorsqu'elle pénètre dans la rétine. Cette lumière a une longueur d'onde comprise entre 400 et 700 nm. Concernant la vision humaine, l'analyse de la couleur est basée sur la colorimétrie ou l'analyse du contenu spectral du stimulus trichromatique détecté par l'oeil; c'est le système XYZ qui décrit des couleurs conformes à ce que perçoit l'Homme. Par ailleurs, dans l'approche physique de la mesure de couleur, la luminance de l'image est représentée par une onde électromagnétique  $E(x, y, \sigma)$ .

## 1.2 Récupération d'images par leur contenu

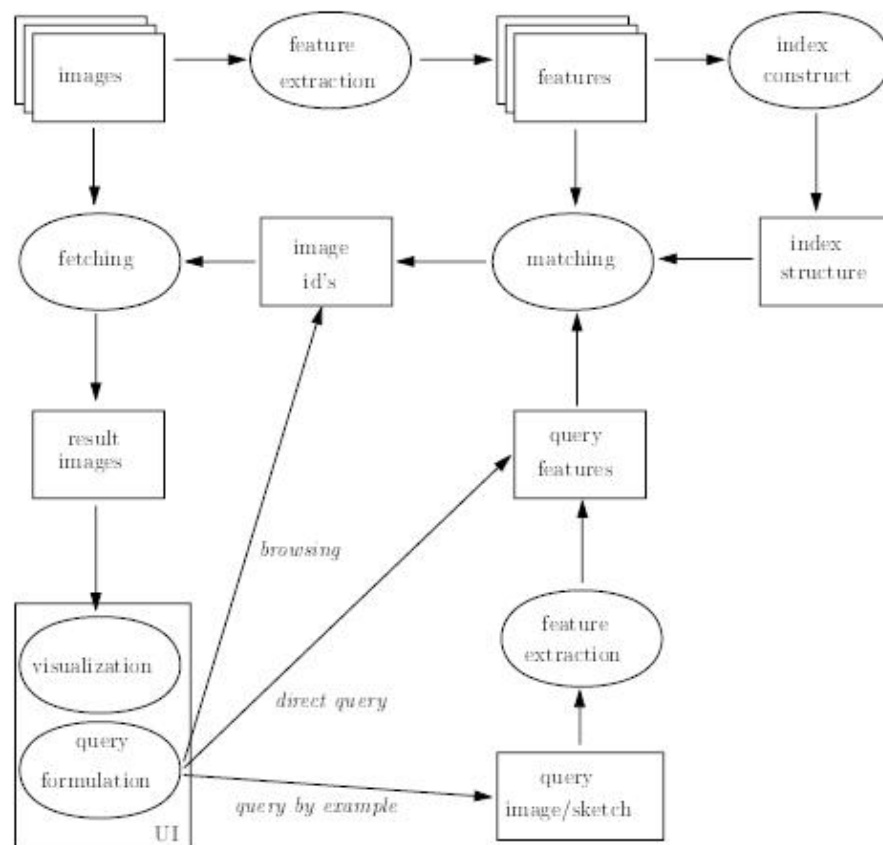


Figure 1.1: Système de récupération d'image par le contenu

## 1. INDEXATION GLOBALE DES IMAGES

---

### 1.2.2 Attribut texture

Dans une image, plusieurs régions sont caractérisées par des structures régulières comparables aux textures qui figurent dans certains vêtements que l'on voit au quotidien. Des chercheurs ont tenté de définir qualitativement la texture dans l'image. Par exemple Pickett considère que la texture est utilisée pour décrire des tableaux de variations de deux dimensions ((Pic70)). Les éléments et les règles d'espacement ou d'arrangement peuvent être manipulés arbitrairement en gardant un indicateur de répétitivité. Pour sa part, Hawkins ((Haw70)) estime que la notion de texture dépend de trois ingrédients:

- un modèle local répété sur une large région par rapport à sa taille ;
- ce modèle consiste en un arrangement non aléatoire de parties élémentaires ;
- les parties sont des entités uniformes ayant appromativement la même dimension partout à l'intérieur de la texture.

Aussi, la texture est souvent caractérisée par la résolution des détails perceptibles dans l'image, qui peuvent être grossière ou fine; on peut parler de taux de grossièreté (coarseness). Par exemple, une pièce de vêtement en laine est plus grossière qu'une pièce de soie sous les mêmes conditions de visualisation. Le caractère plus ou moins grossier de la texture est relié à la période de répétitivité spatiale de la structure locale. Une période grande implique une texture grossière alors qu'une période plus petite implique une texture fine. Toutefois, cette mesure de grossièreté est insuffisante pour mesurer qualitativement la texture. Elle peut orienter les mesures de sorte que l'on puisse prendre les mesures élevées comme une référence à une texture grossière et vice versa.

En outre, la texture est une propriété de voisinage. Sa mesure est donc étroitement liée au voisinage d'observation.

Enfin, la texture artificielle consiste en un arrangement de symboles voire des segments de ligne, des points, des étoiles ou autres, placés sur un arrière plan neutre.

### 1.2.3 Etat de l'art sur les CBIR

Dans ((HSH93)), l' image est segmentée en 8 régions de couleurs homogènes et réduites à 24\*24 pixels. Ces pixels sont directement utilisés dans l'appariement. Pour faire le matching, premièrement une correspondance entre les régions dans la requête et la cible est trouvée. Les régions similaires dans l'autre image sont fusionnées. La similarité de la forme entre deux régions est basée sur le nombre de pixels qui sont superposés. La similarité de couleur entre

### 1.3 Développement d'un descripteur global d'image

---

deux régions est la distance dans l'espace HLS (Hue, Luminosité, Saturation) entre les régions à comparer.

Dans l'approche proposée dans ((CO96)), les couleurs de chaque image sont quantifiées en 13 bins de couleurs. Six valeurs sont associées à chaque bin de couleur

Une autre méthode ((CO96)) consiste à utiliser les attributs couleur et texture, la position et la forme des régions (blobs) et l'arrière plan. La couleur est représentée par 218 bins des coordonnées de couleur dans l'espace  $L^*ab$ . La texture est représentée par le contraste et l'anisotropie dans les régions. La forme est représentée par la surface (excentricité et orientation).

Pour apparier, on utilise une décomposition en couleur singulière (SVD) pour projeter le vecteur d'histogramme en un sous espace de dimension inférieure. Les points résultants sont indexés par un R tree.

Le système ((OM95)) utilise une base de données de description de texte et d'autres éléments textuels comme la date et la position de la photo. Pour chaque image un histogramme de la couleur contenant seulement 20 bins est calculé.

Par ailleurs, dans ((BM00)), les caractéristiques utilisées sont les histogrammes sur la teinte et sur la luminance.

L'appariement consiste à calculer la distance entre deux histogrammes de caractéristiques avec une distance L1, L2 ou Minkovsky. La distance entre deux images est la somme pondérée des distances entre les caractéristiques.

### 1.3 Développement d'un descripteur global d'image

Notre contribution consiste à développer un nouveau descripteur global qui s'inspire du concept de l'MPEG7. Nous combinons les deux attributs couleur et texture en proposant une formule qui mesure à la fois la couleur et la texture. Cela permet une meilleure représentation du contenu de l'image. Pour évaluer notre descripteur, nous avons développé un prototype de système CBIR qui permet de récupérer les images similaires à l'image requête en utilisant une distance appropriée. Notre méthode s'inspire du travail de Geusebroek et al (GdBSG01), où on effectue une segmentation par mesure de la couleur et de la texture par une gaussienne 3D. Nous utilisons le résultat qui montre que la sortie du filtre de Gabor a une nature de Rayleigh pour réduire la dimension de notre vecteur d'attribut.

## 1. INDEXATION GLOBALE DES IMAGES

---

### 1.3.1 Description de l'image par le modèle de couleur Gaussien

Nous avons utilisé dans notre approche les attributs couleur et texture pour décrire l'image. En effet, dans l'approche physique de la mesure de couleur, la luminance de l'image  $L$  est représentée par une onde électromagnétique  $E(x, y, \sigma)$ . Dans la figure 1.1 nous présentons les espaces d'échelle utilisés pour la mesure de la luminance de l'image.

### 1.3.2 Espaces d'échelle

C'est une théorie qui permet de manipuler la structure des images à différentes échelles. Elle introduit en fait le concept de lissage dans lequel une image est un élément dans une famille plus grande qui est celle des images lissées. Elle est gérée par la taille du noyau de lissage utilisé pour supprimer les détails fins. L'espace d'échelle gaussien est le plus utilisé car il permet plusieurs applications. Plusieurs opérateurs visuels peuvent être construits à partir de l'espace d'échelle gaussien. Il permet en plus d'assurer l'invariance à l'échelle nécessaire pour traiter la taille des variations dans les données. Et les espaces d'échelle permettent de gérer la variation de la distance entre un objet et la caméra selon plusieurs échelles.

Pour le cas de signaux 2D  $f(x,y)$ , l'espace d'échelle est un ensemble de signaux définis comme la convolution de  $f(x,y)$  avec les noyaux gaussiens suivants :

$$L(x, y; t) = (g_t * f)(x, y)$$

tel que :

$$g_t(x, y) = \frac{1}{2\pi * t} \exp\left(-\frac{(x^2 + y^2)}{2t}\right)$$

$L$  est l'image  $f$  convoluée avec un espace d'échelle gaussien.

Et  $g_t$  est le noyau gaussien. Le point virgule signifie que la convolution est faite uniquement selon les dimensions  $x$  et  $y$ .  $t$  n'est qu'un paramètre qui représente la variance du noyau gaussien.

Théoriquement, nous définissons une observation comme l'intégral de la densité de l'énergie  $E(x,y)$  et de l'espace d'échelle gaussien. L'intérêt de la gaussienne est d'éviter de supprimer les détails quand l'échelle est grande.

### 1.3.3 Mesure de la couleur: modèle de couleur gaussien de Koendering

Dans ce qui suit nous présentons l'outil de mesure de la couleur qui nous a permis d'avoir notre descripteur.

### 1.3 Développement d'un descripteur global d'image

---

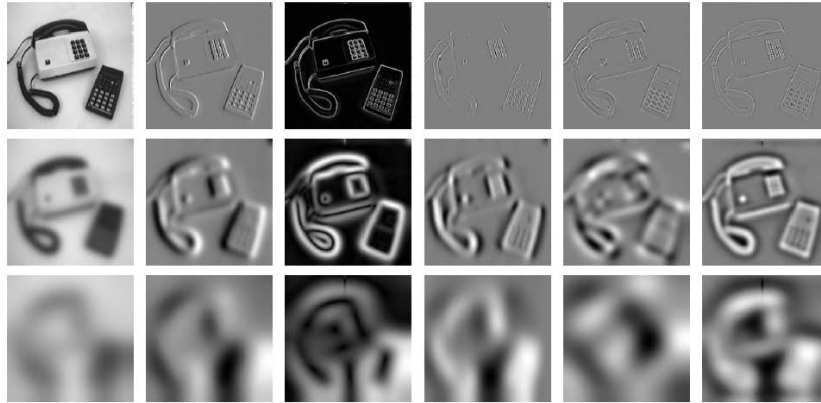


Figure 1.2: Représentation à différents espaces d'échelle

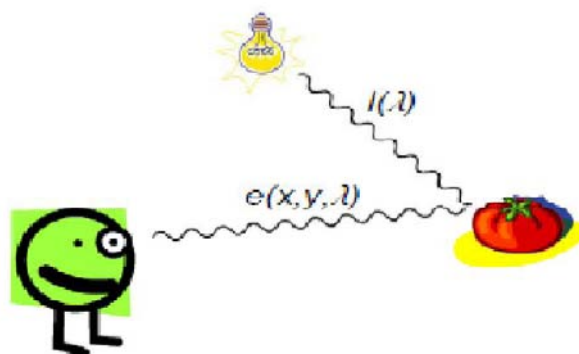


Figure 1.3: Processus de formation d'une image

## 1. INDEXATION GLOBALE DES IMAGES

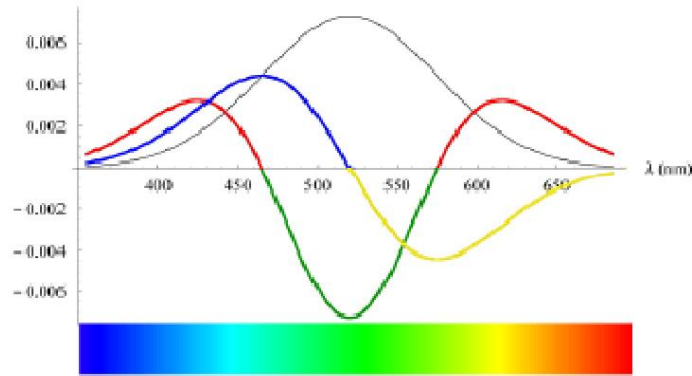


Figure 1.4: Les réponses spectrales du modèle gaussien

L'idée du modèle est que la structure spatiale peut être extraite à partir de données de l'environnement en mesurant des dérivées à un certain ordre. De plus, l'énergie spectrale peut être approximée par le développement de Taylor au point  $\lambda_0$

une mesure physique avec une ouverture est mathématiquement exprimée par :

$$e(x, \lambda) = e(x, \lambda_0) + e_\lambda(x, \lambda_0)(\lambda - \lambda_0) + \frac{1}{2}e_{\lambda\lambda}(x, \lambda_0)(\lambda - \lambda_0)^2 + O((\lambda - \lambda_0)^3)$$

Dans notre cas, la couleur est mesurée par l'intégration de l'énergie électromagnétique sur une bande spectrale. Le modèle de couleur Gaussien est défini par le développement de la distribution de l'énergie au dérivées second ordre. Cette approche reflète le processus de vision humain. Le modèle gaussien mesure les coefficients du modèle de Taylor. La relation entre le modèle de couleur gaussien et celui de l'oeil humain est donné par :

$$\begin{pmatrix} e \\ e_\lambda \\ e_{\lambda\lambda} \end{pmatrix} = \begin{pmatrix} -0.019 & 0.048 & 0.011 \\ 0.019 & 0 & -0.16 \\ 0.047 & -0.052 & 0 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

les dérivées zéro (luminance), premier et second de la fonction gaussienne sont données en fonction de la sensibilité de la longueur d'onde. Selon Koendering, la longueur d'onde centrale est 520nm et l'écart type est de 55 nm.

### 1.3.4 Mesure de la texture

La texture est importante dans la reconnaissance de forme. Daugman (Dau80) a généralisé la fonction de Gabor pour modéliser le champ réceptif dans la vision humaine. Généralement, le filtre de Gabor peut être vu comme la modulation de l'enveloppe Gaussienne et du plan sinusoïdal d'une fréquence et orientation particulière.

### 1.3 Développement d'un descripteur global d'image

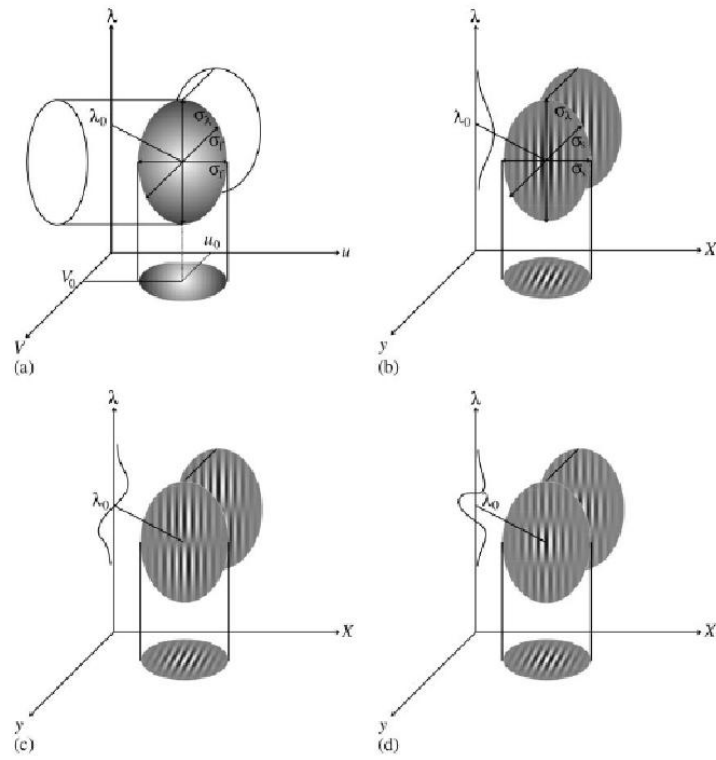


Figure 1.5: Les réponses spectrales du modèle gaussien



## 1. INDEXATION GLOBALE DES IMAGES

---

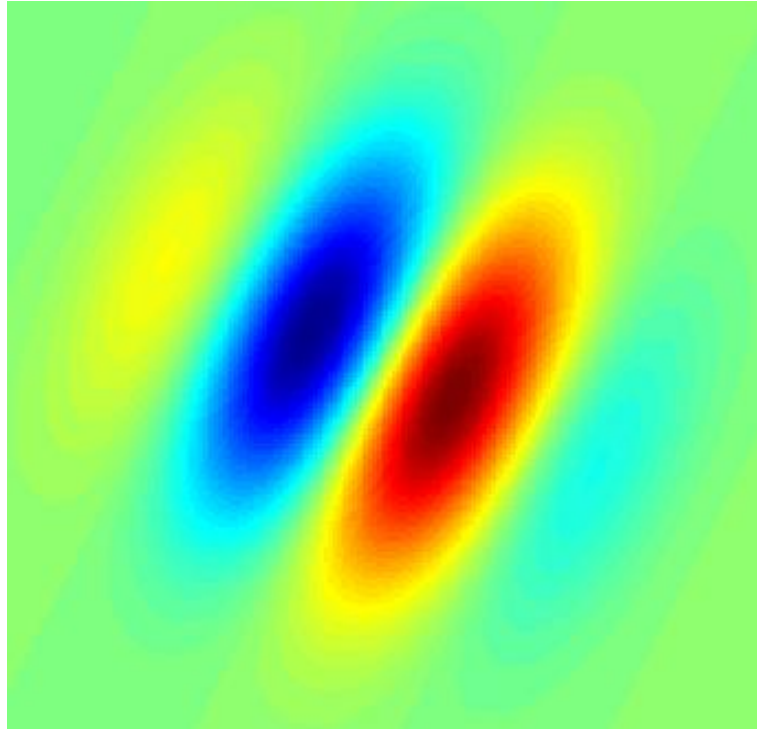


Figure 1.6: Exemple d'un filtre de Gabor à deux dimensions

Un filtre de Gabor est un filtre linéaire utilisé dans le traitement d'images pour la détection de points. Sa réponse impulsionnelle est définie par une fonction harmonique multipliée par une fonction gaussienne. En raison des propriétés de convolution (théorème de convolution), la transformée de Fourier de la réponse impulsionnelle d'un filtre de Gabor est la convolution de la transformée de Fourier de la fonction harmonique et de la transformée de Fourier de la fonction gaussienne.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{(x'^2 + \gamma^2 y'^2)}{\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda}\right) + \psi$$

où

$$x' = x \cos \theta + y \sin \theta$$

et

$$y' = -x \sin \theta + y \cos \theta$$

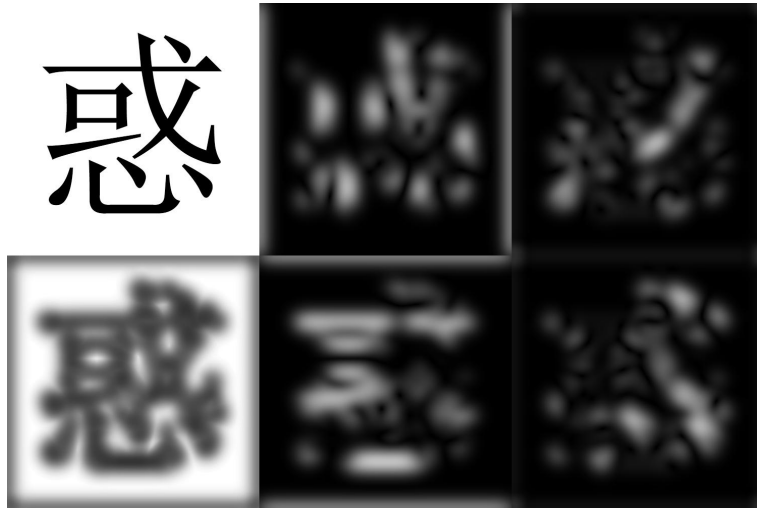


Figure 1.7: Démonstration d'un filtre de Gabor appliqué aux caractères Chinois de l'OCR.. Quatre orientations sont indiquées sur la droite de 0 °, 45 °, 90 ° et 135 °.L'image du caractère original et la superposition de tous les quatre orientations sont indiquées sur la gauche.

### 1.3.5 Mesure conjointe de la couleur et de la texture

Nous allons exprimer en même temps la couleur et la texture. Considérons l'observation

$CT(x, y, \lambda)$

nous avons

$$CT(x, y, \lambda) = \int \int \int E(x, y, \lambda)G(x, y, \lambda, \sigma_s, \sigma_\lambda, \lambda_0)d\lambda \quad (1.1)$$

Avec E est la luminosité et G est un noyau gaussien.

De plus, la texture est caractérisée par sa fréquence spatiale, donc nous travaillons dans l'espace de fréquence avec ses coordonnées  $(u, v, \lambda)$

$$\begin{aligned} CT(u, v) &= \int E(u, v, \lambda)G(u - u_0, v - v_0, \lambda - \lambda_0, \sigma_s, \sigma_\lambda)d\lambda \\ &= G(u - u_0, v - v_0, \sigma_s) \int E(u, v, \lambda)G(\lambda - \lambda_0, \sigma_\lambda)d\lambda(6) \end{aligned}$$

Nous obtenons :

$$CT_n(u, v, \lambda) = G(u - u_0, v - v_0, \sigma_s)$$

## 1. INDEXATION GLOBALE DES IMAGES

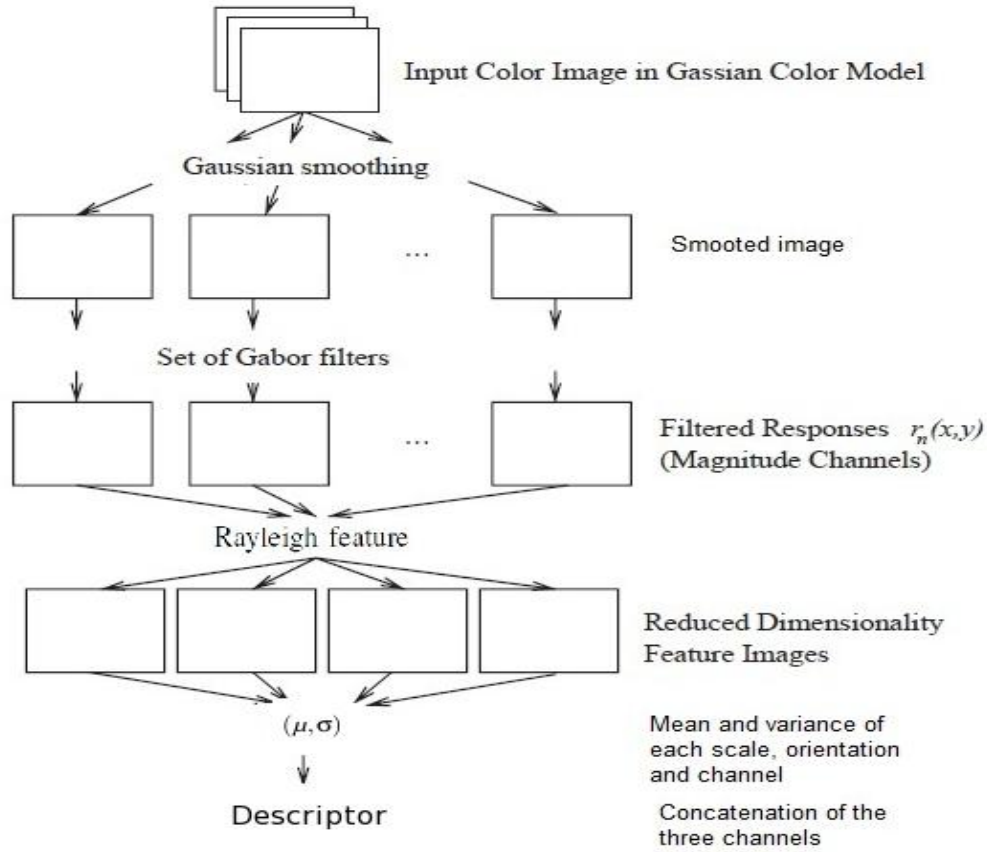


Figure 1.8: Processus de calcul de notre descripteur

$$* \int E(u, v, \lambda) G_n(\lambda - \lambda_0, \sigma_\lambda) d\lambda (7)$$

Une multiplication dans le domaine fréquentiel est équivalente à une convolution dans le domaine spatial. alors

$$CT_n(x, y) = Gb(x, y) * \int E(x, y, \lambda) G_n(\lambda - \lambda_0, \sigma_\lambda) d\lambda (8)$$

où:

$$G_b(x, y) = \frac{1}{2\pi\sigma\beta} e^{-pi \frac{(x-x_0)^2 + (y-y_0)^2}{\sigma_s}} e^{i[\Upsilon_0 x + v_0 y]} (9)$$

$(x_0, y_0)$  est le centre du champ réceptif dans le domaine spatial

$(\Upsilon_0, v_0)$  est la fréquence spatiale optimale du filtre dans le domaine fréquentiel.

$\sigma_s$  est l'écart type de la Gaussienne elliptique selon x et y

## 1.4 Descripteur Couleur Texture

Nous utilisons la formule développée ci-dessus  $CT_n(x, y)$  pour mesurer la couleur et la texture. Pour celà, nous appliquons des ondelettes de Gabor afin d'assurer l'invariance à l'échelle et à l'orientation. La mesure combinée de la couleur et la texture est obtenue par la convolution du filtre de Gabor et de l'image filtrée par la gaussienne.

$$CT_{npq}(x, y) = T_{pq}(x, y) * \int E(x, y, \lambda)G_n(\lambda - \lambda_0, \sigma_\lambda)d\lambda$$

$$CT_{npq}(x, y) = |A_{npq}(x, y) + jB_{npq}(x, y)|(11)$$

Avec n est l'indice du canal de la couleur représentée dans le modèle gaussien. p et q sont les indices de l'orientation et de l'échelle respectivement des ondelettes de Gabor.

L'efficacité de la plateforme couleur texture est affectée par la grande dimension de la sortie du filtre de Gabor.

Pour surmonter ce problème nous avons appliqué le résultat suivant : En effet, Dunn et Higgins (BTM03) ont démontré que la sortie du filtre de Gabor possède une distribution de Rice. Cette propriété est étendue par Bhagavathy (BTM03) au cas d'un filtre de Gabor bi-dimensionnel.

$$A_{npq}(x, y) = R_{npq}(x, y)\cos(\theta_{npq}(x, y))(12)$$

Où  $R_{pqn}(x, y)$  a la PDF suivante :

$$f_R(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + A_0^2}{2 * \sigma^2}\right) I_0\left(\frac{A_0 r}{\sigma^2}\right)$$

et  $I_0(\frac{A_0 r}{\sigma^2})$  représente la fonction de Bessel modifiée du premier ordre.  $R_{pqn}(x, y)$  est une fonction qui contient les informations spatio-spectrales de l'image.

Il a été démontré que la PDF de Rice est comparable à la PDF de Rayleigh pour les petits  $A_0$  et comparable à la PDF Gaussienne pour les grandes  $A_0$ . Le premier cas est plus convenable car il concerne une grande classe de textures.

Bhagavathy, et al ont prouvé qu'il est possible de calculer le descripteur de Rayleigh à partir de Gabor en utilisant l'équation suivante ((BTM03)):

$$\gamma_{mnq}^2 = \frac{1}{2}(\mu_{pqn}^2 + \sigma_{pqn}^2)(13)$$

$\mu_{pqn}, \sigma_{pqn}$  sont respectivement la moyenne et la variance du canal n.

La dimensionalité du filtre de Rayleigh est presque 50% celle du filtre de Gabor avec 3% d'erreur.

## 1. INDEXATION GLOBALE DES IMAGES

---

Principe :

- calculer le descripteur pour l'image texturée gaussienne basée sur la moyenne et la déviation standard pour s échelle, k orientations et 3 canaux ;
- Calculer la caractéristique de Rayleigh en utilisant celle de Gabor pour chaque canal. La dimension du vecteur résultant est  $3*(s*k+2)$  ;
- concatener les 3 descripteurs pour chaque canal

Alors pour chaque canal spectral  $k=0,1,2$ , nous obtenons trois vecteurs descripteurs

$$\begin{aligned}D_{\mu\sigma 0} &= [\mu_{000}, \sigma_{000}, \dots, \mu_{s-1,k-1,0}, \sigma_{s-1,k-1,0}, \mu_{I0}, \sigma_{I0}] \\D_{\mu\sigma 1} &= [\mu_{001}, \sigma_{001}, \dots, \mu_{s-1,k-1,1}, \sigma_{s-1,k-1,1}, \mu_{I1}, \sigma_{I1}] \\D_{\mu\sigma 2} &= [\mu_{002}, \sigma_{002}, \dots, \mu_{s-1,k-1,2}, \sigma_{s-1,k-1,2}, \mu_{I2}, \sigma_{I2}]\end{aligned}\quad (14)$$

Nous formons le descripteur final en concaténant les trois canaux de descripteurs

$$D_{\mu\sigma} = (D_{\mu\sigma 0}, D_{\mu\sigma 1}, D_{\mu\sigma 2})\quad (15)$$

A partir de  $D_{\mu\sigma}$ , le paramètre de Rayleigh de la distribution de la sortie est donné par :

$$\gamma_{mnq}^2 = \frac{1}{2}(\mu_{pqn}^2 + \sigma_{pqn}^2)$$

### 1.4.1 Implémentation du descripteur couleur-texture

Afin d'évaluer notre descripteur dans des tâches de récupération d'images, nous utilisons des caméras RGB. La meilleure transformation linéaire de l'RGB au modèle de couleur Gaussien est donné par :

$$G = MC^T\quad (16)$$

Où

$$M_g = \begin{pmatrix} 0.06 & 0.63 & 0.31 \\ 0.19 & 0.18 & -0.37 \\ 0.22 & -0.44 & 0.06 \end{pmatrix}$$

et

$$C = (RGB)\quad (18)$$

La configuration des ondelettes de Gabor est donnée par (MM96) :

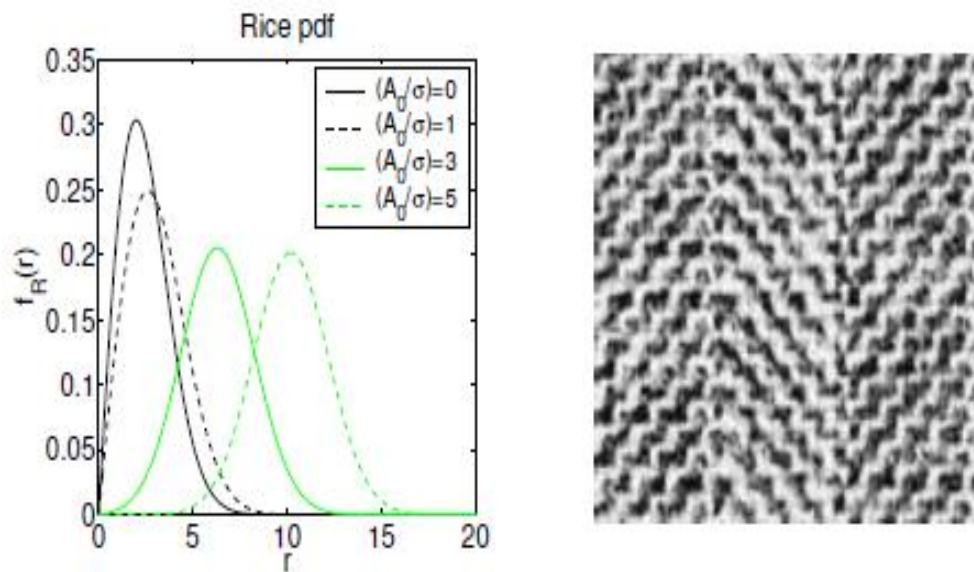


Figure 1.9: Première figure : Rice Pdf, Seconde figure : l'entrée de l'image de texture

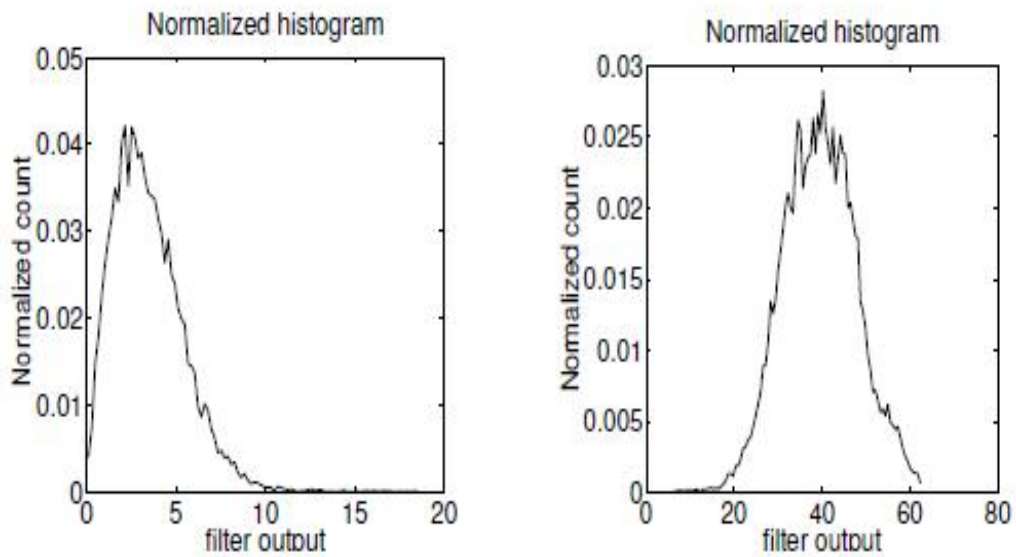


Figure 1.10: Histogramme (avec 100 bins) des sorties du filtre de Gabor pour 2 fréquences centrales

## 1. INDEXATION GLOBALE DES IMAGES

---

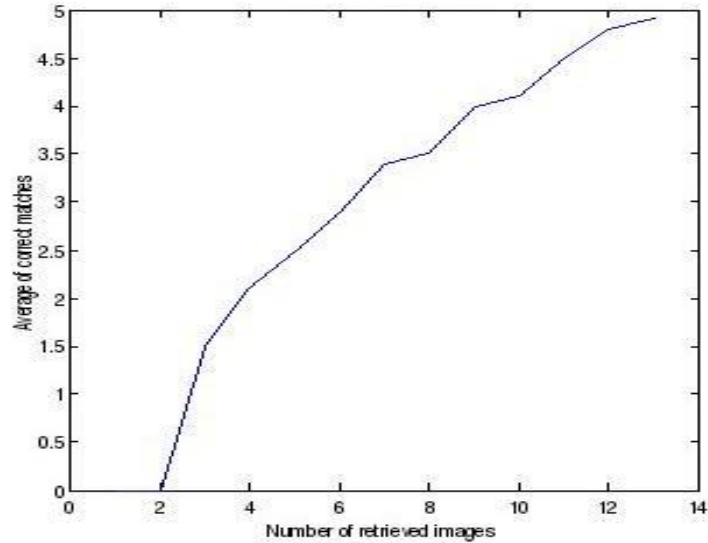


Figure 1.11: Nombre d'images récupérées en fonction des appariements corrects

Echelles	4, 3.5, 2.95, 2.35, 1.75
Fréquences centrales	0.05, 0.08, 0.14, 0.22, 0.33
Orientations	$-\pi/4, -\pi/2, \pi/4, \pi/2$

Les programmes ont été implémentés sous MATLAB. Nous évaluons notre méthode en utilisant 144 images. Elles sont capturées à partir de différents points de vues et sous différentes échelles et orientations. De plus, ils représentent différents thèmes comme la nature, des animaux, des monuments ou des personnes. Chaque image est représentée par un vecteur de 60 éléments ( 5 échelles, 4 orientations et 3 canaux).

Nous prenons un échantillon de 10 images d'une scène particulière. Puis nous appliquons notre descripteur global sur chacune de ces images. Nous effectuons l'appariement en comparant ces descripteurs à celui de l'image requête. Pour cela nous calculons le coefficient de corrélation linéaire entre chaque paire de l'image requête et le vecteur calculé de la base de données images. Le meilleur appariement est obtenu en cherchant le maximum de corrélation. La figure 1.4.1 montre la moyenne des appariements corrects en fonction du nombre des images récupérés. Nous remarquons que cette moyenne croît avec le nombre d'images.

Les images de la figure 1.12 montrent un exemple de simulation :

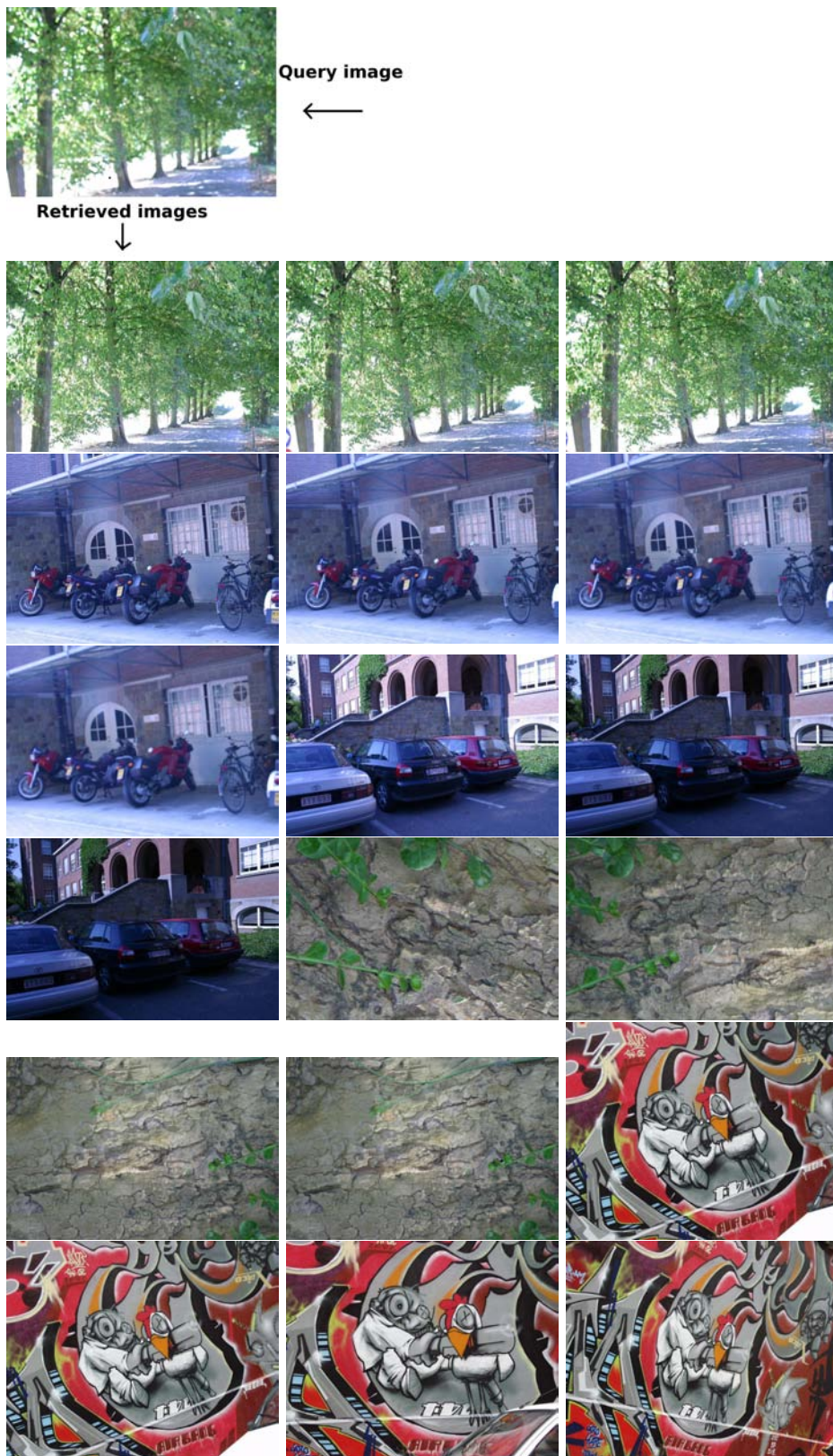


Figure 1.12: Indexation d'images sur indices globaux: validation sur images naturelles.



## 1. INDEXATION GLOBALE DES IMAGES

---

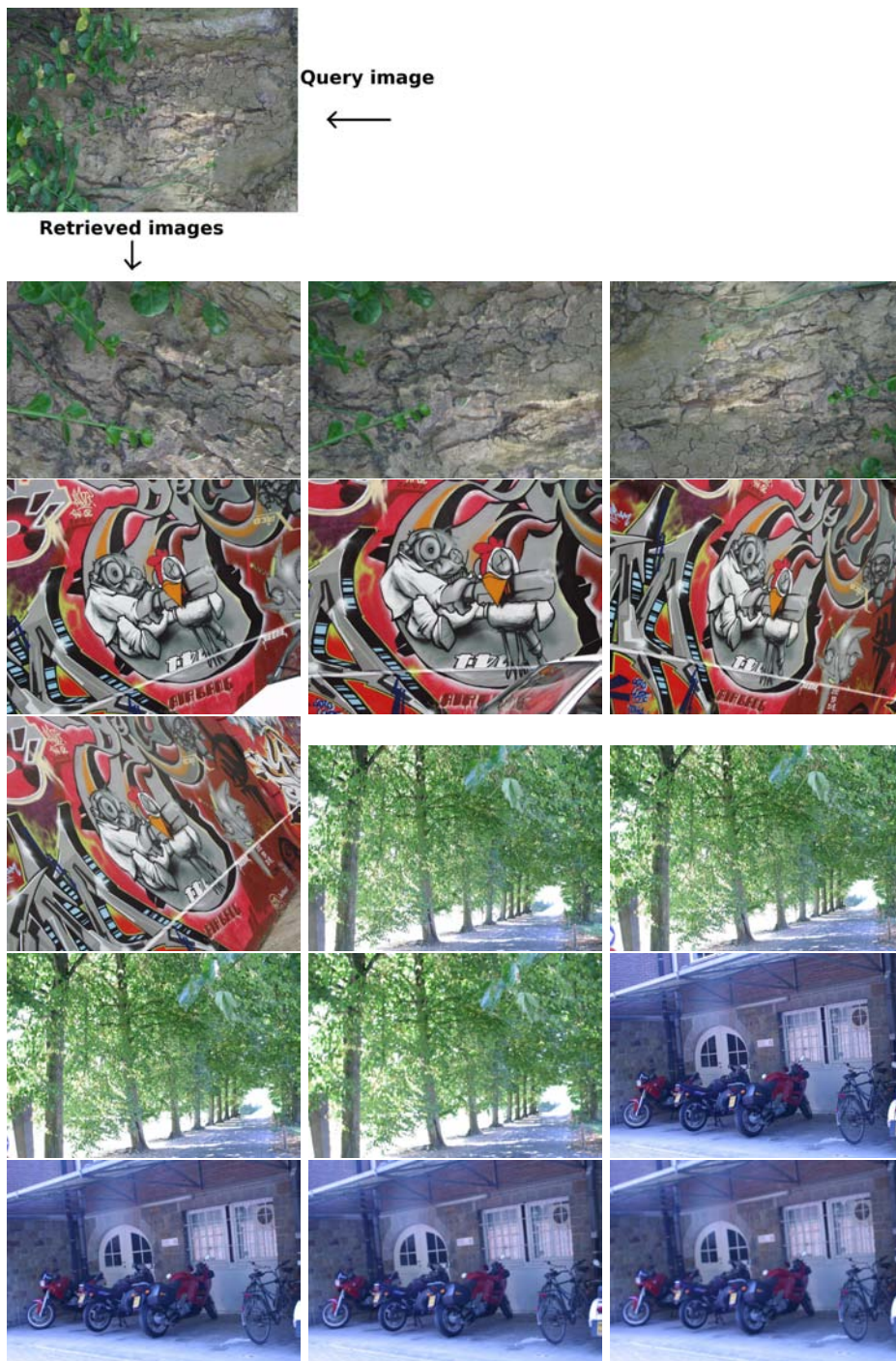


Figure 1.13: Récupération d'image : de l'image la plus similaire à la moins similaire

### 1.5 Conclusion

Dans ce chapitre nous avons présenté un aperçu sur les systèmes actuels de récupération ou indexation d'images. Ces systèmes sont basés sur des descripteurs globaux calculés à partir des attributs couleur, texture et forme. Plusieurs moteurs de recherches d'images déployés sur le web sont basés sur les techniques d'indexation d'images précitées. Cette exploitation des méthodes d'indexation d'images sur le Web, a été rendue nécessaire du fait du contenu important des images et de leur abondance.

Nous avons proposé un index d'image global qui décrit la couleur et la texture globalement dans une image. Notre descripteur est composé d'un vecteur de 60 éléments qui contiennent des mesures statistiques des attributs couleur et texture. Nous avons représenté la couleur par le modèle de couleur gaussien développé par J.Koendering et nous avons mesuré les composantes fréquentielles de la texture par des ondelettes de Gabor. Nous avons ensuite évalué notre index sur une base de données images.

## 1. INDEXATION GLOBALE DES IMAGES

---

# Indexation locale des images

## 2.1 Introduction

Les attributs locaux ou caractéristiques locales (local features) jouent un rôle important dans la reconnaissance d'objets; ces caractéristiques doivent être des éléments de l'image invariants aux différentes transformations comme celles de l'échelle et de l'orientation. La différence de l'approche locale de celle traitée dans le chapitre précédent réside dans l'extraction de caractéristiques qui ici, vont être calculées dans une zone spatiale de l'image. Par conséquent, au lieu d'avoir un seul vecteur qui décrit toute l'image, on obtient une matrice de caractéristiques telle que chaque colonne représente une région de l'image, typiquement le voisinage d'un point d'intérêt. Dans des scènes réelles, les attributs locaux ne doivent pas être affectés par des occultations partielles. Ces attributs doivent aussi être partiellement invariants à l'illumination, aux transformations 3D projectives et aux variations communes de l'objet (échelle et apparence). Ils doivent de plus être suffisamment distinctifs pour permettre d'identifier des objets spécifiques. Les attributs locaux sont fondés sur le principe suivant: ils doivent être des éléments de l'image invariants au processus de formation.

Plusieurs approches permettent de réaliser une invariance aux changements de points de vue. Elles permettent de fournir l'information pertinente contenue dans l'image répondant à plusieurs critères telle que l'invariance et l'optimisation de la complexité du calcul. Deux étapes sont nécessaires pour l'extraction des attributs :

- la construction de régions invariantes de l'image (région support pour le calcul des caractéristiques) ;
- le calcul des descripteurs invariants.

## 2. INDEXATION LOCALE DES IMAGES

---

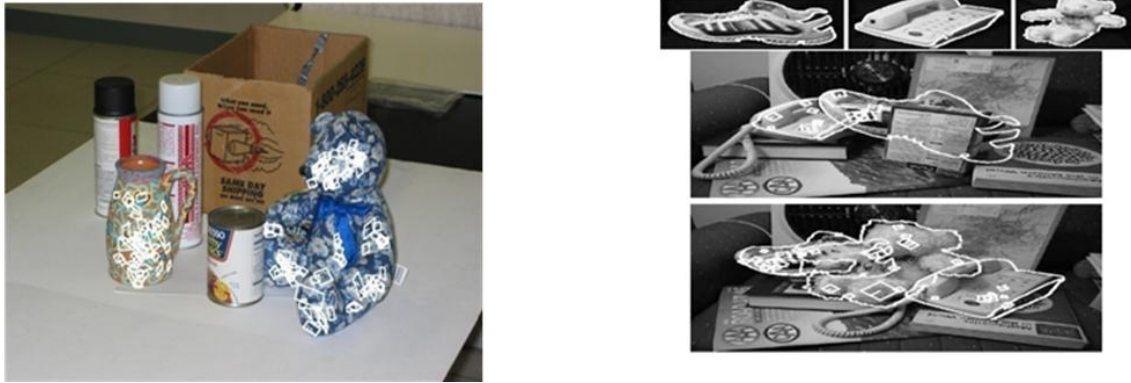


Figure 2.1: Reconnaissance d'objet par appariement de points d'intérêt entre un modèle et une scène

L'un des détecteurs les plus efficaces dans l'indexation par l'apparence ((MS04)) est celui développé dans les années 80 par Harris et Stephens ((HS88)). La version initiale n'était pas invariante à l'échelle et aux transformations affines. Par ailleurs, David Lowe a popularisé le concept de points d'intérêt en développant le détecteur et descripteur appelé *Scale Invariant and Feature Transform (SIFT)* ((Low99)), qui a connu ensuite de très nombreuses améliorations, en particulier avec les points SURF proposés par Bay and al (BTG06).

Nous allons d'abord décrire l'état de l'art avant de proposer notre propre descripteur, largement inspiré par les attributs globaux présentés dans le chapitre précédent. Puis ce descripteur sera validé en l'exploitant pour la reconnaissance d'objets.

## 2.2 Etat de l'art

### 2.2.1 Indexation par apparence

Les approches les plus réputées à ce jour pour la reconnaissance d'objets se fondent sur l'indexation par apparence; voir en particulier les travaux de J.Malik (BMP01) ou M.Hébert (CH02). Un des articles fondateurs de cette approche provient des travaux de Turk et Pentland (TP91), qui l'ont exploitée pour la reconnaissance de visage; Murase et Nayar (MN95) ont repris ces travaux dans un contexte plus général. Ces méthodes traitent le problème de la reconnaissance comme un problème de classification, donc par une approche statistique. Dans cette approche, les objets sont représentés par des vecteurs et la reconnaissance se fonde sur une phase d'apprentissage supervisé; le problème complexe à considérer est celui de la variabilité dans une classe. Par contre, les approches structurelles ou géométriques fondent la reconnaissance d'objets sur les changements de forme selon les points de vue. Elles sont moins dépendantes des informations de luminance, ce qui les rend robustes aux changements

d'illumination.

Nous nous intéressons dans ce chapitre à l'invariance au point de vue, ou tout simplement à *l'invariant* qui permet un mécanisme d'indexation pour des tâches d'appariement; les formes 3D en général n'admettent pas d'invariant. Les invariants combinent donc à la fois des représentations locales de la forme et de l'apparence.

Dans un travail de Jean Ponce (Pon04) des surfaces ont été divisées en petites surfaces considérées comme des points coplanaires. Ensuite il développe une méthode de reconnaissance d'objets où les modèles sont des collections de patches planaires augmentées par leurs invariants et les relations 3D entre eux. Par ailleurs, la méthode de Harris développée en 1988 est probablement la plus connue pour extraire des points d'intérêt. Mais les coins extraits ne sont pas invariants au facteur d'échelle. Lindeberg (Lin94) a introduit le concept de la sélection automatique de l'échelle. On a utilisé des matrices Hessiennes et le Laplacien pour détecter des structures dites BLOB. K. Mikolajczyk et C. Schmid (MS05) ont amélioré ce détecteur pour avoir une plus grande répétabilité. D.Lowe ((Low99)) s'est focalisé sur la rapidité en approximant le LOG ( Laplacian of Gaussian ) par un DOG (Difference of Gaussian). D'autres détecteurs de points d'intérêt ont été développés; citons par exemple celui de Kadir à Oxford (KB01) qui mesure l'entropie dans une région. F.Jurie (JS04) a proposé un détecteur de régions basé sur le contour. Bay and al (BTG06) ont proposé le détecteur SURF, qui est en fait une optimisation du SIFT, donc plus adapté pour des applications temps réel en Robotique. Mikolajczyk et al (HGS05) ont démontré via des études sur les performances que le détecteur qui exploite le hessien de l'image (dérivée d'ordre 2) est plus stable et répétable que celui de Harris.

## 2.2.2 Formulation mathématique

### 2.2.2.1 Changement d'échelle

Un changement d'échelle peut être dû soit à un changement de la distance entre la caméra et l'objet soit à un changement de la longueur focale de l'objectif (dans le cas d'un zoom)(HS88).

#### Invariance au changement d'échelle

Etant donnée une fonction  $f$ , un changement d'échelle  $\alpha$  peut être représenté par un changement de variable:  $f(x) = g(u)$  avec  $g(u) = g(u(x)) = g(\alpha.x)$ . Ceci donne les relations suivantes entre  $f$  et  $g$ .

$$f^{(n)}(x) = \alpha^n g^{(n)}(u) \quad (2.1)$$

## 2. INDEXATION LOCALE DES IMAGES

---

$f^n(x)$  représente la dérivée n-ième de  $f$ .

Cette équation montre que les dérivées énième de  $f$  et de  $g$  sont égales à un facteur multiplicatif  $\alpha^n$  près. Le facteur  $\alpha^n$  peut être éliminé. Des invariants à un changement d'échelle sont données par le rapport suivant :

$$\frac{[f^n(x)]^{\frac{k}{n}}}{f^k(x)}$$

En plus, les invariants d'échelle n'apportent pas de stabilité supplémentaire. Ceci est dû au fait que le calcul numérique est exécuté sur un support. En effet, au cas où les résultats sont calculés par convolution avec les dérivées de la gaussienne, l'équation précédente se réécrit de la manière suivante :

$$\int_{-\infty}^{+\infty} I_1(\vec{x}) G_{i_1 \dots i_n}(\vec{x}\sigma) d(\vec{x}) = \alpha^n \int_{-\infty}^{+\infty} I_2(\vec{x}) G_{i_1 \dots i_2}(\vec{u}\sigma\alpha) d(\vec{u}) \quad (2.2)$$

$G_{i_1 \dots i_n}$  représentent les dérivées de la fonction gaussienne.

Cette équation montre l'importance du support ( $\sigma$  gauche et  $\alpha\sigma$  droit) sur lequel sont effectués les calculs. Ce support doit être adapté au changement d'échelle pour calculer un invariant. Ceci est à l'origine des méthodes multi-échelle.

### Approche multi échelle

Il existe plusieurs approches multi-échelle dans la littérature, en particulier, voir les travaux de K.Mikolajczyk et C.Schmid (MS04). Parmi les premières approches, on trouve celles basées sur les pyramides proposées par Burst et Crowley. Elles utilisent un sous-échantillonnage de l'image. D'autres approches exploitent la notion de l'espace d'échelle où un paramètre continu définit l'échelle. Cette approche a été introduite par (Wit83) et (Koe84). Plus tard, (Lin94) a résumé leur approche. Etant donné que l'espace d'échelle permet de calculer les invariants à une échelle donnée, il est impossible de calculer les invariants à toutes les échelles. Ce qui prouve la nécessité de discrétiser l'échelle, par exemple par des octaves ou des semi-octaves. Avec une telle discrétisation, la caractérisation s'est révélée imprécise et instable. Nous avons choisi de calculer les échelles caractéristiques en utilisant l'algorithme 1, décrit par un pseudo-code.

Dans cet algorithme, nous prenons 4 pixels autour de chaque point d'intérêt. Après nous contruisons une fenêtre autour de chaque point que nous convoluons avec un noyau gaussien. Puis nous extrayons les quatres échelles les plus caractéristiques. Enfin, pour chaque point d'intérêt, nous obtenons 20 échelles (5\*4).

---

**Algorithm 1**

---

- 1: **for**  $i \leftarrow Point_{interet} - 2$  to  $Point_{interet} + 2$  **do**
  - 2:   Création d'une fenêtre autour du point  $i$
  - 3:   Convolution de cette fenêtre avec des Gaussiennes de différentes échelles
  - 4:   Normalisation du résultat
  - 5:   Sélection des 4 valeurs supérieures
  - 6:   Ajout dans le vecteur correspondant au point d'intérêt
  - 7: **end for**
- 

**2.2.2.2 Rotation d'images**

Nous présentons deux approches qui donnent une invariance à la rotation. Dans la première méthode, des mesures différentielles qui sont invariantes aux groupes de déplacements. Elles reposent sur une comcelluleaison de dérivés invariantes aux rotations des images. Dans la deuxième méthode, l'invariance consiste à utiliser des filtres ajustables. Les dérivés sont ajustées dans la direction du gradient et donc invariantes à une rotation image.

**Invariants différentiels**

Ils utilisent le jet local défini ci-après :

Soit  $I$  une image et  $\sigma$  un facteur d'échelle. Le jet local d'ordre  $N$  en un point  $\vec{x}$ , noté  $J^N[I](\vec{x}, \sigma)$ , est défini par :

$$J^N[I](\vec{x}, \sigma) = L_{i_1, \dots, i_n}(\vec{x}, \sigma) | (\vec{x}, \sigma) \in I * R^+, n = 0, \dots, N \quad (2.3)$$

$L_{i_1, \dots, i_n}$  est la drive n-ième de l'image par rapport aux variables  $i_k$  ( $k=1\dots n$ ) obtenue par la convolution de  $I$  avec la différentielle  $G_{i_1, \dots, i_n}(\vec{x}, \sigma)$  de la fonction Gaussienne. :

$$L_{i_1, \dots, i_n}(\vec{x}, \sigma) = G_{i_1, \dots, i_n}(\vec{x}, \sigma) * I(\vec{x}) \quad (2.4)$$

$G(\vec{x}, \sigma)$  et  $G_{i_1, \dots, i_n}$  sont définis par les équations :

$$G(\vec{x}, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\vec{x}^2}{2\sigma^2}\right) \quad (2.5)$$

et

$$G_{i_1, \dots, i_n} = \frac{d^n}{d_{i_1} \dots d_{i_n}} G(\vec{x}, \sigma), n = 0 \dots N \quad (2.6)$$

En utilisant le jet local, Koendering et Romeny (KvD87) suggèrent de calculer des invariants pour le groupe de déplacements  $SO(2)$ . Ce qui exige le calcul des dérivées de manière stable pour pouvoir calculer ces invariants à un ordre élevé. Les invariants différentiels sont



## 2. INDEXATION LOCALE DES IMAGES

---

utilisés jusqu'au troisième ordre. Ils sont regroupés dans un vecteur noté  $\vec{V}$ . La première partie de ce vecteur est constitué d'un ensemble complet et irréductible d'invariants différentiels jusqu'au deuxième ordre.

$$\vec{V} = \begin{bmatrix} L \\ L_i L_j \\ L_i L_{ij} L_j \\ L_{ii} \\ L_{ij} L_{ji} \end{bmatrix} = \begin{bmatrix} L \\ L_x L_x + L_y L_y \\ L_{xx} L_x L_x + 2L_{xy} L_x L_y + L_{yy} L_y L_y \\ L_{xx} + L_{yy} \\ L_{xx} L_{xx} + 2L_{xy} L_{xy} + L_{yy} L_{yy} \end{bmatrix} \quad (2.7)$$

Les  $L_i$  représentent des éléments du jet local.  $L$  est par exemple la fonction de la luminance convoluée avec une gaussienne. La formulation de cette première partie du vecteur est donnée en notation d'Einstein et en coordonnées cartésiennes. En notation cartésienne, les indices  $x$  et  $y$  représentent respectivement la dérivation par rapport aux variables  $x$  et  $y$ , par exemple  $L_{xy} = \frac{d^2}{dx dy} L$ . En notation d'Einstein, un indice  $i$  représente la sommation des dérivations par rapport à l'ensemble des variables :

$$L = \sum_i L_i = L_x + L_y \quad (2.8)$$

et

$$L = L_{xx} + L_{xy} + L_{yx} + L_{yy} \quad (2.9)$$

La deuxième composante de ce vecteur est la magnitude du gradient et la quatrième, le Laplacien. Le calcul des invariants peut être effectué pour différentes tailles  $\sigma$  de la gaussienne car ils sont définis à partir des  $L_i$ . Quant à la deuxième partie du vecteur, elle est constituée d'un ensemble complet d'invariants du troisième ordre. En notation d'Einstein, ces invariants sont :

$$\vec{V}[5..8] = \begin{bmatrix} \epsilon_{ij}(L_{jkl} L_i L_k L_l - L_{jkk} L_i L_l L_l) \\ L_{ii j} L_j L_j L_k L_k - L_{ijk} L_i L_j L_k \\ -\epsilon_{ij} L_{jkl} L_i L_k L_l \\ L_{ijk} L_i L_j L_k \end{bmatrix} \quad (2.10)$$

$\epsilon_{ij}$  représente le tenseur canonique anti-symétrique. Soit,  $\epsilon_{12} = -\epsilon_{21} = 1$  et  $\epsilon_{11} = -\epsilon_{22} = 0$ .

### Jet ajustable

Il est possible de calculer les dérivées dans une direction donnée à partir du jet local défini. Pour être invariant à la rotation, cette direction peut être celle du gradient. Nous présentons les formules pour calculer jusqu'à l'ordre 3 les dérivées dans une direction  $\theta$  donnée. Les  $L_x$ ,

$L_y, \dots$  sont les éléments du jet local.

$$L'(\theta) = L_x \cos(\theta) + L_y \sin(\theta) \quad (2.11)$$

$$L''(\theta) = L_{xx} \cos^2(\theta) + 2L_{xy} \sin(\theta) \cos(\theta) + L_{yy} \sin^2(\theta) \quad (2.12)$$

$$L'''(\theta) = L_{xxx} \cos^3(\theta) + 3L_{xxy} \cos^2(\theta) \sin(\theta) + 3L_{xyy} \sin^2(\theta) \cos(\theta) + L_{yyy} \sin^3(\theta) \quad (2.13)$$

la dérivée d'ordre  $n$  dans une direction donnée dépend des  $n + 1$  dérivées d'ordre  $n$ . Afin de représenter de façon complète l'ensemble des dérivées dans un ordre donné  $n$ , il faut utiliser  $n + 1$  dérivées directionnelles correspondant à  $n + 1$  directions  $\theta_{n,i}$   $i=0,n$ . Pour assurer la stabilité, on doit espacer régulièrement les  $n+1$  directions  $\theta_{n,i}$ . Pour obtenir les dérivées indépendantes de la rotation existante entre deux images, la direction  $\theta_{i,0}$  doit de plus être rapportée à l'image. Si cette direction correspond à la direction du gradient, les orientations sont alors :

$$\theta_{n,i} = \frac{i\pi}{n+1} + \theta_g$$

.

et

$$\theta_g = \arctg\left(\frac{L_y}{L_x}\right)$$

.

Cependant cette direction  $\theta_g$  présente une source d'instabilité des méthodes utilisant les jets ajustables.

### Normalisation en taille de l'image

Nous notons que la rectangularité des pixels introduit une anisotropie qui fausse le calcul des dérivées. Alors, la forme des pixels doit être carrée pour obtenir effectivement des dérivées invariants à la rotation. Il faut donc normaliser l'image. Ceci est fait par l'intermédiaire d'une interpolation linéaire sur les colonnes de l'image en utilisant un facteur de réduction égal au facteur  $\alpha_v/\alpha_u$ . Ce dernier représente le ratio entre la largeur et la longueur d'un pixel. Des expérimentations ont montré que ce facteur est stable et dépend peu du calibrage.

## 2. INDEXATION LOCALE DES IMAGES

---

### 2.2.3 Le détecteur de Moravec

Le principe du détecteur développé en 1977 par Moravec (HS88) est de considérer une petite fenêtre et de calculer la moyenne de variations d'intensité dans cette fenêtre.

Trois cas sont distingués :

- Les changements sont dus à de petites variations de l'intensité si le patch de la fenêtre est plat. Ainsi, si la fenêtre est intersectée avec un bord, le changement le long de la frontière sera petit.
- Dans la direction perpendiculaire, il sera grand.
- Si le patch est un coin ou un point isolé, tous les décalages seront grands. Alors, le coin sera détecté si le changement du minimum par n'importe quel décalage est large.

Le changement produit par le déplacement (x;y) est :

$$S_{x,y} = \sum_{uv} [E_{x+u,y+v} - E_{u,v}]^2 = \sum_{u,v} W_{u,v} [xX + yY + o(x^2, y^2)]^2 \quad (2.14)$$

$$X = E * (-1, 0, 1) = \frac{dE}{dx} \quad (2.15)$$

$$Y = E * (-1, 0, 1)^T = \frac{dE}{dy} \quad (2.16)$$

S est le moment second ordre.

#### -Quelques inconvénients de l'opérateur Moravec (HS88)

- La réponse est anisotropique

A cause des orientations discrètes, seulement un changement discret tous les 45 degré est effectué. Les petits changements possibles peuvent être réalisés en considérant un développement analytique à l'origine du changement.

$$S_{x,y} = \sum_{uv} [E_{x+u,y+v} - E_{u,v}]^2 = \sum_{u,v} W_{u,v} [xX + yY + o(x^2, y^2)]^2 \quad (2.17)$$

$$X = E * (-1, 0, 1) = \frac{dE}{dx} \quad (2.18)$$

$$Y = E * (-1, 0, 1)^T = \frac{dE}{dy} \quad (2.19)$$

Pour les petits changements,

$$S(x, y) = Ax^2 + 2Cxy + By^2 \quad (2.20)$$

$$A = X^2 * W$$

$$B = Y^2 * W$$

$$C = (XY) * W$$

- La réponse est bruitée

Parce que la fenêtre est cellulaire et rectangulaire, il est indispensable d'utiliser une fenêtre de lissage. Cette dernière est donnée par :

$$W_{u,v} = \exp\left(-\frac{u^2+v^2}{2\sigma^2}\right) \quad (2.21)$$

- L'opérateur répond rapidement aux variations de contours

Seulement le minimum de l'énergie E est pris en considération.

$$S(x, y) = (x, y)M(x, y)^T \quad (2.22)$$

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (2.23)$$

E(x,y) est similaire à une fonction d'autocorrélation. Le M présente sa forme à l'origine.

Considérons  $\alpha$  et  $\beta$  les valeurs propres de M :

Soient :

$$Tr(M) = \alpha + \beta = A + B$$

$$Det(M) = \alpha.\beta = AB - C^2$$

et

$$R = Det - kTr^2$$

- Si la courbure est petite, la fonction de l'autocorrélation est plate. Donc la région de l'image dans la fenêtre a approximativement une intensité constante.

## 2. INDEXATION LOCALE DES IMAGES

---

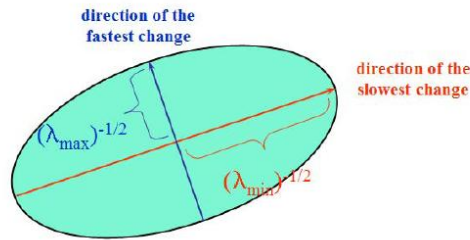


Figure 2.2: ellipse d'incertitude correspondant à une analyse des valeurs propres de la matrice d'autocorrélation

- Si l'une est grande et l'autre est petite, la fonction de l'autocorrélation est " ridge shaped "
- Si les courbures sont grandes, ceci indique un coin

### 2.2.4 Détecteur de Harris et ses variantes

#### Détecteur de Harris Laplace :

Nous présentons le détecteur de Harris Laplace et Harris Affine ((MS04)) invariants aux transformations de l'échelle et aux transformations affines. En résumé, le détecteur de Harris Laplace utilise l'échelle adaptée pour localiser des points dans l'espace de l'échelle. Donc il sélectionne les points pour lesquels le Laplacien de la Gaussienne atteint un maximum sur une échelle donnée, en utilisant l'équation suivante:

$$|LoG(x, \sigma_n)| = \sigma_n^2 |L_{xx}(x, \sigma_n) + L_{yy}(x, \sigma_n)| \quad (2.24)$$

Avec  $\sigma_n$  est l'échelle.

La fonction de Harris multi échelle est utilisée pour sélectionner les positions des points d'intérêt. On applique la formule d'échelle suivante :

$$\sigma_n = \xi^n \sigma_0 \quad (2.25)$$

qui donnent les différentes échelles en fonction de  $\sigma_0$  qui permettent de calculer les caractéristiques selon chacune des échelles.

$\xi$  est le facteur de l'échelle entre niveaux consécutifs.

Lindeberg (MS04) propose  $\xi = 1.4$ .

En chaque niveau de la représentation, les points d'intérêt sont extraits en détectant les maximums locaux dans les 8 voisinages d'un point  $x$ .

Le moment second ordre  $M(x, \sigma_n)$  est calculé en utilisant l'échelle d'intégration  $\sigma_I$  et l'échelle de différentiation  $\sigma_D$  tel que

$$\sigma_I = \sigma_D = \sigma_n$$

Pour chaque point, un algorithme itératif détecte simultanément l'emplacement et l'échelle du point d'intérêt. L'échelle du point d'intérêt est calculée en utilisant la fonction LoG. Ainsi, les points pour lesquels la réponse de LoG n'atteint aucun extremum et pour lesquels la réponse est au-dessous du seuil sont rejetés.

**-Algorithmes pour itération :**

Soit le point initial  $x$  et l'échelle  $\sigma_I$

1. pour les points  $x^k$  trouver l'extremum local à une échelle donnée. Sinon, rejeter les points. Les échelles sont obtenues par :  $\sigma_I^{k+1} = t\sigma_I^k$
2. Pour sélectionner les points  $\sigma_I^{k+1}$ , détecter l'emplacement spatial  $x^{k+1}$  d'un maximum de la mesure de Harris
3. revenir à 1 si  $\sigma_I^{k+1} \neq \sigma_I^k$  et  $x^{k+1} \neq (x^k)$

**Détecteur de Harris-Affine :**

La matrice second ordre est souvent utilisée pour décrire la structure de l'image locale. Par ailleurs, pour rendre cette matrice indépendante de la résolution de l'image, on utilise une matrice adaptée aux changements d'échelle.

$$M(x, y, \sigma_I, \sigma_D) = \sigma_D g(\sigma_I) \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix} \quad (2.26)$$

$\sigma_I$  : l'échelle de l'intégration

$\sigma_D$  l'échelle de la différentiation

$L_a$ : La dérivée calculée dans la direction

$g$  : est un noyau gaussien.

Les dérivées locales sont calculées avec un noyau gaussien de la dimension déterminée par l'échelle locale.

Les valeurs propres de cette matrice présentent deux principaux changements du signal dans le voisinage d'un point. Cette propriété permet l'extraction de points pour lesquels les

## 2. INDEXATION LOCALE DES IMAGES

---

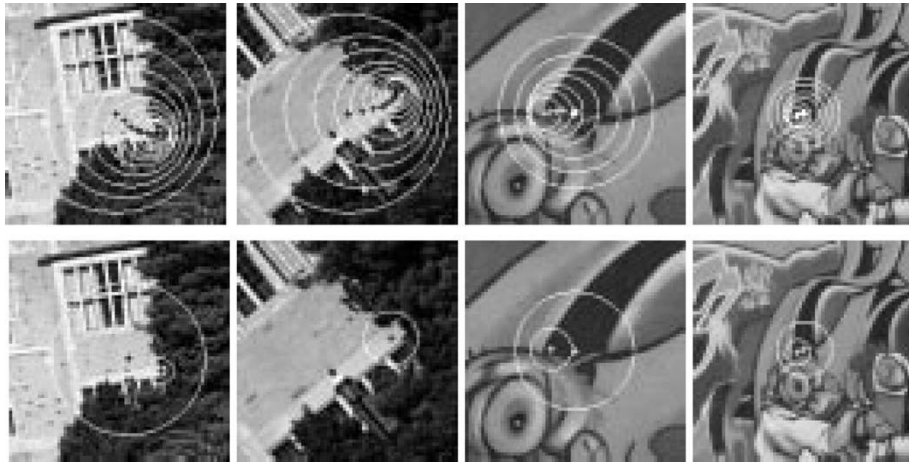


Figure 2.3: Détection de points d'intérêt invariants à l'échelle. Les réponses spectrales du modèle gaussien. En haut, la sélection manuelle de points de Harris multi-échelle correspondant à une structure locale. En bas, les points d'intérêt sélectionnée par l'approche de Harris-Laplace

deux courbures sont considérables.

$$\text{Cornerness} = \det(M(x, y, \sigma_I, \sigma_D)) - \alpha \cdot \text{trace}^2(M(x, y, \sigma_I, \sigma_D)) \quad (2.27)$$

Les maximums locaux de la cornerness déterminent l'emplacement des points d'intérêt.

### - Sélection automatique de l'échelle :

La sélection de l'échelle automatique était largement étudiée par Lindeberg ([Lin94](#)). L'idée dans la sélection des attributs est relative à une fonction donnée qui atteint un extremum.

Les points initiaux sont détectés avec un détecteur de Harris multi échelle avec un changement large entre deux échelles successives de 1.4 qui correspondent au gap entre deux niveaux d'espaces d'échelles voisins.

Le détecteur de Harris Affine est initialisé avec les points d'intérêt extraits par le détecteur de Harris multi échelle. Le détecteur de Harris permet de déterminer la localisation spatiale des points d'intérêt. La matrice de la forme pour chaque point d'intérêt est obtenue par le calcul du descripteur de second ordre avec des échelles d'intégration et de différenciation ( $\sigma_D = s * \sigma_I$ ) pour limiter l'espace de recherche.

### Matrice d'adaptation de la forme

La méthode utilisée généralement pour l'adaptation de la forme itérative est effectuée dans le domaine de transformation de l'image. L'image est transformée puis un noyau circulaire est appliqué à la place du noyau Gaussien Affine. Ceci permet une implémentation récursive du filtre gaussien pour calculer  $L_x$  et  $L_y$ . Une fenêtre locale  $w$  est centrée au point d'intérêt

x et transformée par la matrice suivante dans l'étape k de l'algorithme itératif.

$$U = \prod_k (M^{\frac{-1}{2}})^{(k)} U^{(0)} \quad (2.28)$$

A chaque itération, la matrice M est calculée de nouveau. La plus grande valeur propre est fixée  $\lambda_{max}(u) = 1$ . Ce qui signifie que le patch de l'image est élargi dans la direction  $\lambda_u$ . les échelles sont sélectionnées automatiquement dans chaque itération. Pour chaque point, les échelles d'intégration et de différentiation déterminent le moment du second ordre.

### **L'échelle d'intégration ((MS04)):**

L'échelle caractéristique est sélectionnée pour chaque point. Pour préserver l'invariance aux changements de la taille, on sélectionne l'échelle d'intégration  $\sigma_i$  où le Laplacien normalisé atteint un maximum local. S'il y a une grande déformation affine, le changement d'échelle est très différent sur chaque direction.

Il est important de sélectionner l'échelle d'intégration dans chaque itération après l'application de la transformée U. Pour cela, une procédure similaire à celle de Harris Laplace est utilisée. Les points initiaux convergent vers un point où l'échelle et la matrice du moment second ordre ne changent pas.

### **L'échelle de différentiation :**

L'échelle locale de différentiation utilise l'échelle d'intégration et la mesure d'isotropie Q. L'échelle locale a une influence sur la convergence de la matrice du moment second ordre. La procédure itérative converge vers une matrice ayant des valeurs propres égales. Il est à noter que la mesure de Harris sélectionne les points qui possèdent deux valeurs propres larges. Une grande différence entre les valeurs propres conduit à une grande échelle dans une direction avec la transformation U. Dans ce cas les points ne convergent pas vers une solution stable à cause du bruit. La sélection de l'échelle locale permet d'avoir un ratio de valeurs propres raisonnable. Les points ne convergent pas à une solution stable à cause du bruit. Par ailleurs, l'échelle de différentiation locale peut être fixée proportionnellement à l'échelle d'intégration  $\sigma_d = s * \sigma_i$  où s est un facteur constant. Ceci accélère significativement les itérations mais il reste que quelques points ne convergent pas à cause de la grande différence entre les valeurs propres.

### **Harris Laplace simplifié :**

Cette simplification accélère la détection des points d'intérêt. Les points initiaux sont détectés avec un détecteur de Harris multi échelle. L'espace de l'échelle est construit avec la fonction de Harris. D'autres fonctions existent. Les maximums locaux à chaque niveau de l'échelle sont détectés. La vérification est faite en calculant la fonction LoG. Si la réponse de



## 2. INDEXATION LOCALE DES IMAGES

---

LoG n'atteint aucun extremum ou la réponse est au-dessous d'un seuil, cela permet d'obtenir un ensemble de points caractéristiques avec les balances associées.

L'approche de Harris Laplace donne un ensemble de représentations compactes de points caractéristiques dans l'image et la dimension de l'échelle. La première approche donne une précision meilleure de la position et de l'échelle du point d'intérêt. La seconde est un compromis entre la précision et la complexité de calcul.

### 2.2.5 Détecteur de régions saillantes

Le point clé dans l'approche de Kadir et Brady ((KB01)) est que les régions saillantes de l'image montrent une "imprévision" ou une "surprise" dans leurs attributs locaux et sur l'échelle spatiale.

Ainsi, trois étapes sont considérées :

- calcul de l'entropie de Shanon des attributs locaux de l'image. On considère l'intensité ou la couleur sur un rang d'échelles  $HD(s)$
- Sélection des échelles dont l'entropie sur la fonction de l'échelle montre un pic (SP)
- Calcul du changement d'amplitude de la PDF en tant que fonction d'échelle sur chaque pic ( $WD(s)$ )

La saillance finale est obtenue par le calcul du produit de  $HD(s)$  et  $WD(s)$  sur chaque pic. A l'intérieur d'une fenêtre circulaire de rayon  $s$ , l'histogramme de valeur de pixels est utilisé pour estimer la PDF locale. Les étapes 1 et 3 mesurent l'espace de attributs et la prédiction de l'inter-échelle en même temps. L'étape 2 sélectionne les échelles optimales.

La prédiction de la région est déterminé par l'entropie des attributs locaux. L'entropie des intensités des pixels montre le degré de changement. Dans le cas où l'entropie est grande, alors l'imprédiction est grande. Sinon, elle est petite. Ensuite, les échelles sur lesquelles l'entropie à un pic sont recherchées.

Les points dans lesquels l'entropie a un pic sont utiles dans le sens où ils sont covariants avec une invariance isotropique d'échelle, aux rotations et aux translations. Aussi, considérons la figure 2.4; dans l'illustration sur l'échelle de l'entropie, la permutation de chaque patch local n'affecte pas son entropie. Trois permutations sont utilisées, une rampe linéaire, un réordonnement aléatoire et un gradient radial. Dans l'étape 3, la mesure de l'imprédiction inter-échelle  $W_d$  prend la valeur de l'entropie de telle sorte que quelques permutations sont préférées à d'autres.

Dans le cas continu, la mesure est définie par :

$$y_d(s_p, x) = h_d(s_p, x)w_d(s_p, x) \quad (2.29)$$

$$h_d(s_p, x) = - \int p(I, s, x) \log_2 p(I, s, x) dI \quad (2.30)$$

$p(I, s, x)$  : est la densité de probabilité de la fonction de l'échelle I et de la position s.

$$w_d(s, x) = s \int \left| \frac{d}{ds} p(I, s, x) \right| dI \quad (2.31)$$

D'autres attributs tels que la couleur ou l'orientation peuvent être utilisé également. Cette approche offre un modèle plus général de saillance de attributs et d'échelle comparée aux techniques de détection conventionnelles.

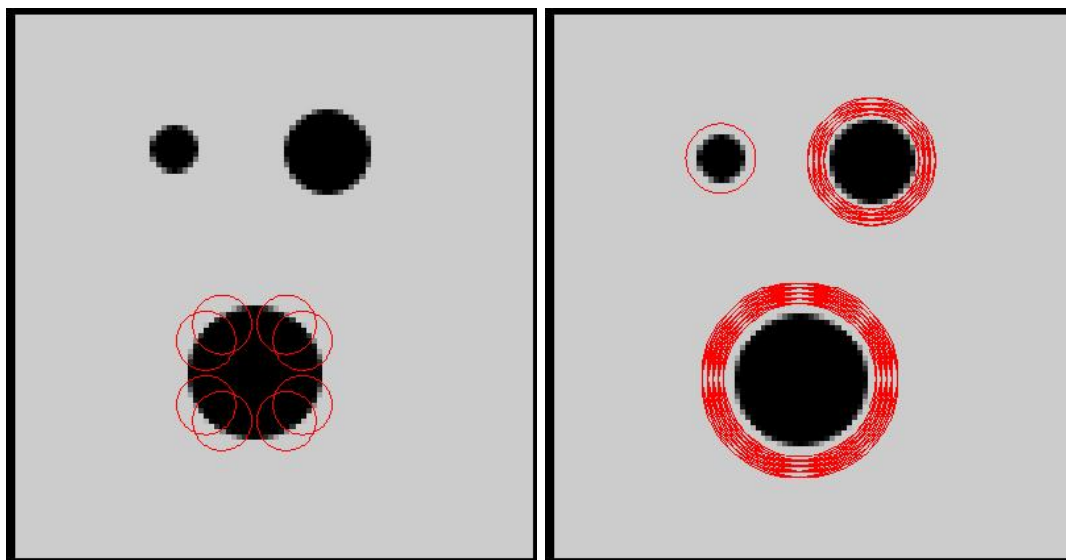


Figure 2.4: Les parties les plus saillantes sont sélectionnées (gauche) en utilisant l'entropie et (droite) en utilisant les pics dans l'entropie versus l'échelle pondérée par la somme des différences absolues au pic(droite)

### saillance affine invariante

Il est extrêmement important que la méthode soit totalement invariante aux transformations géométriques affines. Ceci peut être effectué en remplaçant la fenêtre d'échantillonnage circulaire par une ellipse. En effet, sous les transformations affines, les cercles deviennent des ellipses.

Pour cela, le paramètre d'échelle est remplacé par le vecteur  $s = (s, \eta, \theta)$ ,  $\eta$  étant le quotient des axes,  $\theta$  est l'orientation et les axes majeurs et mineurs sont donné par  $\frac{s}{\sqrt{\rho}}$  et  $s\sqrt{\rho}$ . La

## 2. INDEXATION LOCALE DES IMAGES

---

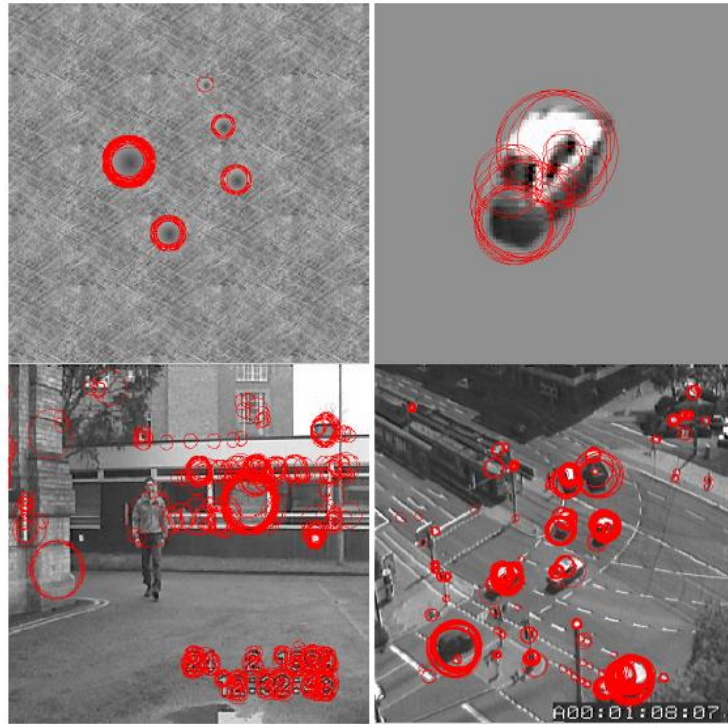


Figure 2.5: Détection des points les plus saillants et leurs échelles

dégénérescence peut être créée lorsque la dimension de la fenêtre d'échantillonnage augmente.

### 2.2.6 Filtres orientables (Steerable filters)

Un banc de filtre est normalement utilisé pour représenter des patches de textures. Pour caractériser la variabilité de l'apparence de la texture, les statistiques jointes ou marginales des descripteurs sont utilisées. Il est courant d'utiliser un prototype de filtres qui dépend de l'échelle et des rotations ((FA91)). Si le noyau du filtre est bien localisé dans les domaines spatial et fréquentiel, ceci permet un échantillonnage du spectre local le long des axes ayant un sens sémantique par rapport à l'échelle et à l'orientation. Pour une échelle donnée  $\sigma$ , soit  $l(x, y, \theta)$  le composant du descripteur au pixel  $(x, y)$  qui correspond au noyau et l'orientation  $\theta$ . Donc, après une rotation autour de  $(x, y)$ , le descripteur  $l(x, y, \theta)$  devient  $l(x, y, \theta - \delta\theta)$ .

L'orientation (steerability) est utile surtout pour extraire des invariants à la rotation. C'est la propriété d'une fonction qui effectue une transformation sous une action du groupe de Lie.

Considérons des filtres orientables tels que les fonctions de base soient des rotations entre  $0$  et  $\pi$  du noyau prototype  $h(x, y)$ .

$h_i = l(x, y, \theta_i)$  est la version filtrée de  $l(x, y)$ . Pour un angle générique  $\theta$ , nous avons :

$$l(x, y, \theta) = \sum_{i=0}^{N-1} l_i(x, y) k_i(\theta) \quad (2.32)$$

$k(\theta)$  : fonction d'interpolation indépendante l.

$h_i(x, y)$  : une version du noyau avec une rotation  $\frac{i\pi}{N}$

La rotation par  $\theta$  est vue comme une combinaison des fonctions  $h_i(x, y)$  avec des coefficients qui dépendent uniquement de  $\theta$ .

Si nous considérons  $H_M(x)$  un polynôme hermitique d'ordre M, M est l'ordre de la dérivée et la fonction est  $f(x, y) = (-1)^M H_M(x) G(x, y)$ .

$N=M+1$  sont suffisantes pour synthétiser  $H_M(x)G(x, y)$

Les fonctions d'interpolation sont trigonométriques et polynomiales.

$$k_i(\theta) = \frac{2}{M+1} \sum_{n=0}^{\frac{M-1}{2}} \cos(2n+1)(\theta - \theta_i) \quad (2.33)$$

$$k_i(\theta) = \frac{1}{M+1} \left( 1 + 2 \sum_{n=1}^{\frac{M}{2}} \cos(2n)(\theta - \theta_i) \right) \quad (2.34)$$

### 2.2.7 Descripteur SIFT

#### La détection des extrema de l'espace d'échelle (Low04):

Les points clé (keypoints) sont détectés avec des filtres en cascade qui utilisent des algorithmes efficaces pour identifier les positions candidates. Cette première étape consiste à identifier les positions et les échelles qui peuvent être assignés d'une manière répétitive sur différentes vues du même objet. La recherche des caractéristiques stables permet de détecter les positions invariantes aux changements d'échelles. Cela est fait en utilisant la fonction continue d'espace d'échelle. L'espace d'échelle d'une image peut être défini comme la résultante de la convolution de cette image avec des noyaux gaussiens. Pour une détection efficace des positions des points clé stables, Lowe (Low04) a proposé l'utilisation des extremums des espaces d'échelle dans la fonction différence de gaussienne convoluée avec l'image  $D(x, y, \sigma)$ . L'image initiale est incrémentalement convoluée avec des gaussiennes pour produire des images séparées par un facteur constant k dans l'espace d'échelle. Chaque octave de l'espace d'échelle est divisé en un nombre entier s d'intervalles  $k = 2^{\frac{1}{s}}$ . On produit s+3 images dans la pile des images lisses pour chaque octave. Une fois l'octave traitée, nous rééchantillons l'image gaussienne ayant le double de l'échelle initiale en prenant chaque deuxième pixel dans

## 2. INDEXATION LOCALE DES IMAGES

---

chaque ligne et colonne. La précision de l'échantillonnage relatif  $\sigma$  n'est pas différente de celle du début de l'octave précédente et le calcul est très réduit.

### La détection des extrema locaux

Pour détecter les maxima et les minima locaux de  $D(x, y, \sigma)$ , chaque point est comparé à ses 8 voisins dans l'échelle au dessous et au dessus. S'il est supérieur ou inférieur à tous ces voisins il est sélectionné. Le coût de cette vérification est raisonnablement faible puisque la plupart des échantillons seront éliminés après la première vérification. La détection de la fréquence d'échantillonnage dans les domaines de l'image et de l'échelle est très intéressante. Cependant, il n'y a pas d'espace minimum d'échantillon pour détecter tous les extrema puisque ces extrema peuvent être proches d'une manière arbitraire les uns des autres. Donc nous devons chercher une solution qui permet l'efficacité et la complémentarité.

### localisation des keypoints

Après avoir détecté les keypoints candidats, l'étape suivante est de faire une interpolation détaillée des données pour avoir la position, l'échelle et le ratio de la courbure principale. Cette information permet de rejeter quelques points ayant un faible contraste ou qui ne sont pas bien localisés sur le contour. L'implémentation initiale de cette approche localise simplement les points clé sur la position et l'échelle du point échantillon central. Mais une méthode est utilisée pour interpoler une fonction quadratique 3D aux points échantillons locaux afin de déterminer la position interpolée du maximum. Ses expériences ont montré une amélioration d'appariement et de stabilité. L'approche de Brown et Lowe (Low04) utilise le développement de Taylor (jusqu'au terme quadratique) de la fonction d'espace d'échelle  $D(x, y, \sigma)$ .

$$D(X) = D + \frac{dD^T}{dX}X + \frac{1}{2}x^T \frac{d^2D}{dX^2}x \quad (2.35)$$

D et ses dérivées sont évaluées au point échantillon et  $X = (x, y, \sigma)$  est l'offset de ce point. La position de l'extremum  $\hat{x}$  est déterminée en prenant les dérivées de la fonction selon X.

$$\hat{X} = -\frac{d^2D^{-1}}{dX^2} \frac{dD}{dX} \quad (2.36)$$

Brown estime que la Hessian et les dérivées de D doivent être approximés en utilisant les différences de D.

**Elimination des réponses des contours** Pour garantir la stabilité, ce n'est pas suffisant de rejeter seulement les points clé avec un mauvais contraste. En effet, la différence des fonctions gaussiennes aura une réponse forte le long des contours même si la position n'est pas bien déterminée et donc instable aux petites variations du bruit. Donc, un pic défini dans la fonction de différence de gaussienne aura une courbure principale large au long du contour

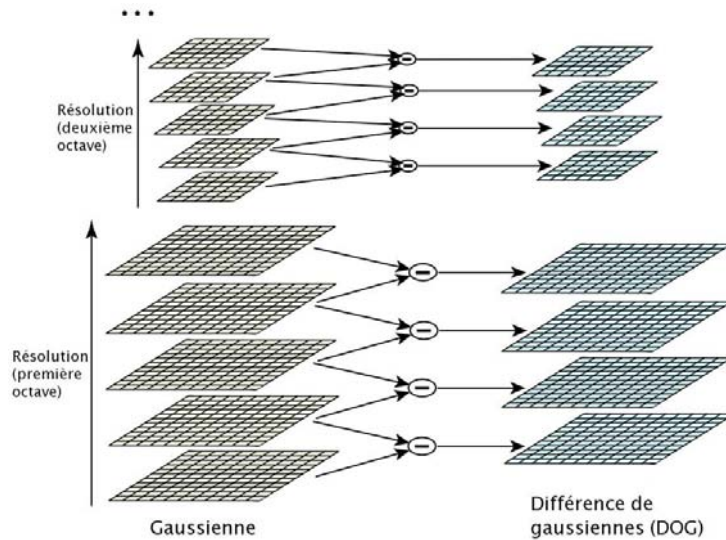


Figure 2.6: Détection des caractéristique dans l'espace d'échelle en utilisant une pyramide de gaussienne (à gauche) les niveaux voisins d'une pyramide gaussienne sub-octave sont soustraites pour produire des images de différences de gaussienne.

mais une petite dans la direction principale. Une matrice Hessienne  $2 \times 2$  permet de calculer la courbure dans une position et une échelle du points clé :

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \quad (2.37)$$

Les dérivées sont estimées en prenant les différences des points échantillonnés voisins. Les valeurs propres de  $H$  sont proportionnelles à la courbure principale de  $D$ . En s'inspirant de l'approche utilisée par Harris et Stephen (HS88), on peut explicitement éviter le calcul des valeurs propres puisque cela dépend uniquement du ratio.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (2.38)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (2.39)$$

Si le déterminant est négatif, alors la courbure a plusieurs signes et donc elle est écartée (non maximal). Soit  $r$  le quotient entre les magnitudes ayant la plus large et la plus petite valeur propre:

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (2.40)$$

## 2. INDEXATION LOCALE DES IMAGES

---

qui dépend uniquement du ratio des valeurs propres. Pour vérifier que le quotient des courbures principales est inférieur à un seuil  $r$ , on vérifie que :

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (2.41)$$

Soit  $r=10$  qui élimine les points clé ayant un ratio supérieur à 10.

### Les orientations

En affectant une orientation à chaque points clé basée sur les propriétés locales de l'image, nous pouvons assurer une invariance à la rotation. Une autre approche (SM97) a été développée dans laquelle chaque propriété de l'image est basée sur des mesures invariantes à la rotation. Cependant, cette approche limite les descripteurs qui peuvent être utilisés. Dans l'approche de SIFT, l'échelle du point clé est utilisée pour sélectionner l'image lissée par une gaussienne en utilisant les plus proches échelles; Ceci étant, pour que tous les calculs soient faits avec une invariance à l'échelle. Pour chaque échantillon de l'image  $L(x,y)$  sur cette échelle, la magnitude du gradient  $m(x,y)$  et l'orientation  $\theta(x,y)$  sont calculées en utilisant des différences de pixels :

$$m(x,y) = ((L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2)^{\frac{1}{2}} \quad (2.42)$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \quad (2.43)$$

L'histogramme des orientations est formé à partir des orientations du gradient des points échantillons à l'intérieur de la région autour du point clé. L'histogramme des orientations a 36 cellules couvrant 360 degrés d'orientation. chaque échantillon additionné à l'histogramme est pondéré par sa magnitude du gradient et par une fenêtre circulaire avec une échelle de 1.5 fois celle de l'échelle du point clé. Les pics dans l'histogramme des orientations correspondent aux directions d'orientation du gradient local. Les pics les plus grands dans l'histogramme sont détectés et après, chaque pic local à l'intérieur de 80% du plus grand pic est aussi utilisé pour créer un point clé avec cette orientation. Ainsi, avec des positions multiples au pics de magnitudes similaires, on aura de multiples points clé créés à la même position et échelle avec différentes orientations. Seulement 15% de points sont assignés à de multiples orientations. Mais ils contribuent clairement à la stabilité de l'appariement. Finalement, la parabole est interpolée aux trois valeurs de l'histogramme proches de chaque pic pour interpoler la position des pics afin d'avoir une précision meilleure.

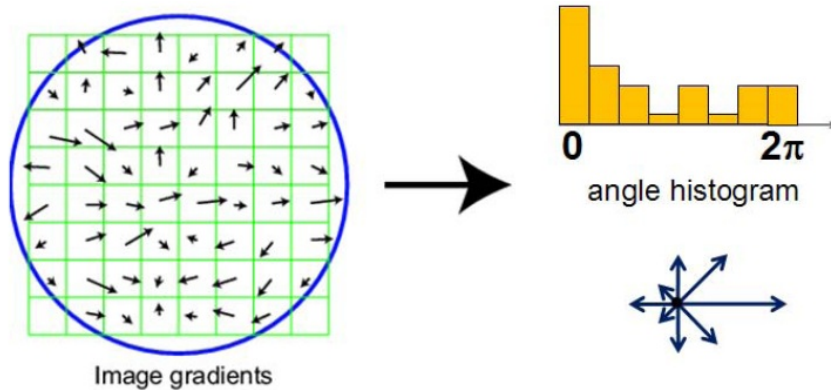


Figure 2.7: Une estimée d'orientation dominante peut être calculée en créant un histogramme de toutes les orientations du gradient ( pondérées par leurs amplitudes après le seuillage des petites valeurs) et ensuite en trouvant les pics significatifs dans cette distribution ((Low04))

### Le descripteur local de l'image :

Les opérations suivantes ont assigné une position, une échelle et une orientation à chaque point clé. Ces paramètres exigent une répétabilité pour décrire la région locale de l'image, et ainsi produire une invariance à ces paramètres. Ensuite il faut calculer un descripteur pour la région locale de l'image et puis produire une invariance à ces paramètres.

On utilise le concept des neurones complexes dans le cortex visuel primaire. Ces neurones complexes répondent au gradient d'une orientation particulière et une fréquence spatiale mais la position du gradient dans la rétine est amené à être décalé sur un petit champ réceptif . Edelman et al (EIP97) ont posé l'hypothèse que la fonction de ces neurones complexes consiste à faire l'appariement et la reconnaissance des objets 3D à partir d'un ensemble de points de vues.

### La représentation du descripteur

La magnitude du gradient de l'image et l'orientation sont échantillonnées autour de la position du point clé en utilisant l'échelle du point clé pour sélectionner le niveau du lissage gaussien. Pour assurer une invariance à l'orientation, les coordonnées du descripteur et les orientations du gradient sont déviées relativement à l'orientation du point clé. Pour l'efficacité, les gradients sont précalculés pour tous les niveaux de la pyramide. Une fonction gaussienne de pondération avec  $1.5x$ (largeur de la fenêtre de description) est utilisée pour affecter un poids à l'amplitude de chaque point échantillon. Malgré le fait que le poids diminue de manière lisse, l'objectif de cette fenêtre gaussienne est d'éviter les changements qui peuvent accompagner les changements petits dans la position, et de donner moins d'importance aux gradients qui sont loins du centre du descripteur.



## 2. INDEXATION LOCALE DES IMAGES

---

Le descripteur du point clé permet des décalages significatifs dans la position du gradient en créant des histogrammes d'orientation sur 4x4 échantillons de régions. La figure 2.7 montre 8 directions pour chaque histogramme d'orientation avec la longueur de chaque flèche correspondante à l'amplitude de cette entrée de l'histogramme. Un échantillon de gradient gauche peut être décalé plus de 4 positions d'échantillon en contribuant au même histogramme à droite. Il est aussi important d'éviter les effets des frontières dans lesquels le descripteur change subitement. Le descripteur est formé à partir d'un vecteur contenant les valeurs de toutes les entrées de l'histogramme d'orientation correspondant aux longueurs des flèches de la partie droite (figure 2.7). La figure montre un tableau 2x2 des histogrammes d'orientation. Les expériences montrent que les meilleurs résultats sont obtenus avec un tableau 4x4 et des histogrammes de 8 cellules d'orientation pour chaque point clé. Finalement, le vecteur des caractéristiques est modifié pour réduire l'effet des changements d'illumination. Le vecteur est normalisé à la longueur unité. Un changement dans le contraste de l'image dans lequel chaque valeur de pixel est multipliée par une constante va multiplier les gradients par cette même constante. Ainsi, ce changement de contraste sera évité par la normalisation du vecteur. Le changement de la luminosité dans laquelle une constante est ajoutée à chaque pixel de l'image n'affectera pas les valeurs du gradient puisqu'elles sont calculées à partir de la différence de pixel. Le descripteur sera alors invariant aux changements affines dans l'illumination.

Nous retrouvons plusieurs applications industrielles des SIFT :

- Assistance des conducteurs de voitures
- La photographie digitale
- Le suivi des visages
- Les films et les vidéos
- Les jeux et la reconnaissance des gestes.
- Industrie électronique
- Reconnaissance d'objets pour les engins mobiles

## 2.3 Développement d'un descripteur couleur invariant à l'échelle et à la rotation

### 2.3.1 Nature de notre détecteur-descripteur

Notre objectif est de développer un détecteur et un descripteur ayant une dimensionalité basse tout en assurant une haute performance. Beaucoup de descripteurs ont été développés dans la littérature afin de résoudre des problèmes comme la reconnaissance d'objet et la navigation de robots. Nous démontrons que notre descripteur est distinctif et répétable dans la reconnaissance d'objets. De plus, l'invariance aux déformations géométriques et photométriques est d'une grande importance pour un descripteur. Pour cela, nous avons assuré l'invariance à la rotation et à l'échelle. Ces éléments offrent en effet un compromis entre la complexité du point d'intérêt et sa robustesse aux déformations. Ainsi, nous avons considéré les effets second ordre, comme les échelles anisotropiques et les effets perspectifs. En ce qui concerne les effets photométriques, nous avons utilisé le modèle de couleur gaussien pour représenter la luminance

### 2.3.2 Développement mathématique

Nous avons présenté dans la section précédente le détecteur de Harris. En effet, ce détecteur est efficace en termes de performance. Il est basé sur le calcul de la matrice d'autocorrélation ce qui le rend convenable au calcul des structures locales dans les images. Nous l'adaptions à l'échelle pour qu'il soit invariant à la résolution de l'image. Pour cela, nous appliquons le moment second ordre adapté à l'échelle :

$$M(x, y, \sigma_I, \sigma_D) = \sigma_D g(\sigma_I) \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix} \quad (2.44)$$

Les dérivations locales sont calculées par le noyau gaussien déterminé par l'échelle  $\sigma_D$ . On effectue après le moyennage des dérivées dans le voisinage du point par un lissage avec une gaussienne de taille  $\sigma_I$ . Cependant le détecteur de Harris est adapté seulement aux images de niveau de gris. Montesinos ((MGD98)) l'a généralisé aux images couleur. Les points d'intérêt produits par leurs détecteurs sont définis comme les extremas locaux positifs de l'image à niveaux de gris intermédiaires.

$$R(x, y) = \det(M(x, y)) - k(\text{trace}(M(x, y)))^2 \quad (2.45)$$

## 2. INDEXATION LOCALE DES IMAGES

---

k vaut 0.04 et  $M(x,y)$  est la matrice 2x2 construite à chaque point de l'image par l'intermédiaire des images niveaux de gris.

$$\begin{pmatrix} M_{11}(x, y) & M_{12}(x, y) \\ M_{12}(x, y) & M_{22}(x, y) \end{pmatrix} \quad (2.46)$$

$M_{11}(x, y)$ ,  $M_{12}(x, y)$  et  $M_{22}(x, y)$  sont définies par :

$$M_{11}(x, y) = G_{\sigma_i} \otimes (r_x^2 + g_x^2 + b_x^2) \quad (2.47)$$

$$M_{12}(x, y) = G_{\sigma_i} \otimes (r_x r_y + g_x g_y + b_x b_y) \quad (2.48)$$

$$M_{22}(x, y) = G_{\sigma_i} \otimes (r_y^2 + g_y^2 + b_y^2) \quad (2.49)$$

$G_{\sigma_i}$  est une gaussienne 2D isotropique avec une variance  $\sigma_i^2$ .  $\otimes$  est l'opérateur de convolution et  $c_x$  et  $c_y$  représentent les dérivées gaussiennes premier ordre du canal  $c$  de l'image d'origine  $I(x,y)$  avec  $c \in r,g,b$ . Ces dérivées sont implémentées en utilisant une gaussienne 1D avec une variance  $\sigma_D^2$ .

Dan notre travail, nous proposons d'utiliser le modèle de couleur gaussien développé par Koendering. Nous rappelons que ce modèle (décrit dans le chapitre 1) calcule les coefficients  $E^{\lambda_0, \sigma_\lambda}, E_\lambda^{\lambda_0, \sigma_\lambda}, E_{\lambda\lambda}^{\lambda_0, \sigma_\lambda}$  de l'expansion de Taylor de la distribution spectrale de l'énergie pondérée par une gaussienne  $\lambda_0$  et l'échelle  $\sigma_\lambda$ . étant l'énergie électromagnétique de l'image. La matrice du modèle est donnée par J.M.Geusebroek dans (GdBSG01): un pseudo-code décrit notre méthode dans l'algorithme 2.

$$M_g = \begin{pmatrix} 0.06 & 0.63 & 0.31 \\ 0.19 & 0.18 & -0.37 \\ 0.22 & -0.44 & 0.06 \end{pmatrix} \quad (2.50)$$

### 2.3.2.1 Calcul d'un vecteur d'échelles caractéristiques

Nous voulons sélectionner les échelles caractéristiques de la structure locale déterminée par notre point d'intérêt. Ces échelles identifient la partie dans laquelle il y a une similarité entre l'opérateur de détection de l'échelle et la structure locale de l'image. Nous proposons l'algorithme suivant pour le calcul des échelles caractéristiques :

1. pour  $i = \text{Feature Point}-2, \text{Feature Point}+2$
2. Créer une fenêtre autour du point  $i$

## 2.3 Développement d'un descripteur couleur invariant à l'échelle et à la rotation

---

**Algorithm 2**

---

```
1: in ← Lire image
2: R ← in(1)
3: B ← in(2)
4: G ← in(3)
5: M ←  $\begin{pmatrix} 0.06 & 0.63 & 0.31 \\ 0.19 & 0.18 & -0.37 \\ 0.22 & -0.44 & 0.06 \end{pmatrix}$ 
6: Diagonaliser M → (α,β,γ)
7: S ← α.R + β.G + γ.B
8: Lxx ← Gxx.S
9: Lxy ← Gxy.S
10: Lyy ← Gyy.S
11: M(x, y, σI, σD) ← σDg(σI)  $\begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix}$ 
```

---

3. Convoluer cette fenetre avec des gaussiennes de différentes echelles
4. Calculer la norme scalaire du résultat
5. Prendre les 4 plus grandes échelles
6. Concaténer dans le vecteur qui correspond au point d'intérêt
7. fin pour

### 2.3.2.2 Calcul de l'orientation

Pour assurer l'invariance à la rotation, nous calculons l'orientation des points d'intérêt. Nous allons comcelluler les informations locales de notre détecteur et la distribution du gradient relative. Ainsi, pour chaque échelle nous extrayons les échelles caractéristiques. La formule est donnée par :

$$R_{ij} = a \tan(E_{ij} - E_{i+1,j}, E_{i,j+1} - E_{ij})$$

. Où E est la valeur du pixel. Nous concaténons les différentes orientations dans le même vecteur.

### 2.3.3 Calcul du descripteur

Nous calculons des caractéristiques de la texture en utilisant des ondelettes de Gabor à différentes orientations. Autour de chaque point d'intérêt calculé par notre variante du détecteur

## 2. INDEXATION LOCALE DES IMAGES

---

de Harris multi-échelle, nous effectuons une convolution avec un noyau de Gabor ((Dau85)) :

$$T_{p,q}(x, y) = a^{-p}t(x', y'), a > 1, p, q \in Z \quad (2.51)$$

où

$$x' = a^{-p}(x \cos(\theta) + y \sin(\theta)) \quad (2.52)$$

$$y' = a^{-p}(-x \sin(\theta) + y \cos(\theta)) \quad (2.53)$$

où

$$\theta = \frac{q\pi}{K}$$

K est le nombre d'orientations

Finalement, notre descripteur est illustré par le tableau suivant :

x y	$\sigma$	r	t
espace	échelle	orientation	texture

Figure 2.8: Structure de notre détecteur-descripteur, 2 champs pour la position spatiale, 21 pour l'échelle, 5 pour l'orientation, 9 pour la texture.

## 2.4 Reconnaissance d'objets

### 2.4.1 Généralités

La reconnaissance d'objets dans une scène est une des plus anciennes tâches dans le domaine de la vision par ordinateur et reste une des plus difficiles. Chaque technique de reconnaissance des formes est directement liée au décryptage des informations contenues dans l'environnement naturel. Au cours de la dernière décennie, des efforts remarquables ont été faits pour construire de nouveaux systèmes de vision capables de reconnaître des objets dans des environnements encombrés.

Nous utilisons une méthode géométrique d'appariement entre une image et une base d'images. Les données sont extraites préalablement des images. Cette méthode utilise des données symboliques alors qu'elle calcule des grandeurs numériques.

Un certain nombre d'approches sont basées sur le paradigme suivant : calcul d'hypothèses et vérification. Après avoir extrait les caractéristiques à partir des images à reconnaître, on associe aux caractéristiques des modèles 2D contenus dans une base d'images. Cependant,

la recherche exhaustive de tous les modèles existants dans la base nécessite un temps de calcul polynomial. Plusieurs recherches se sont focalisé sur la diminution de la complexité des appariements soit en utilisant une évaluation récursive d'hypothèses ou des méthodes d'indexation et de hachage pour obtenir une accélération significative. Pour l'indexation, la mise en correspondance des caractéristiques et la recherche d'un modèle de la base sont remplacées par un mécanisme de "look-up table". Par ailleurs, le groupe Vision d'Oxford a utilisé des invariants projectifs comme élément d'indexation.

Notre méthode est basée sur la transformée de Hough. Elle choisit le modèle en recherchant un point d'accumulation dans l'espace des transformations ((Bal81)). Dans l'approche de Lowe ((Low99)), chaque point d'intérêt est associé à son plus proche voisin en utilisant une distance euclidienne. La transformée de Hough est utilisée pour prédire l'orientation et l'échelle à partir des hypothèses d'appariement.

### Construction séquentielle du modèle

Notre modèle est construit à partir d'indices visuels regroupés en vues - clés. Il est construit séquentiellement à partir d'un ensemble d'images obtenues par la focalisation de la caméra sur plusieurs vues de l'objet.

### Initialisation du modèle associé au proto-objet

Nous commençons par une comparaison de la première image de la séquence avec la base de modèles existants. Si le modèle existe déjà, il est associé au proto-objet courant et sera utilisé pour la suite du traitement de la séquence. Sinon, un nouveau modèle est créé tel que la première image est la première vue clé.

### Mise en correspondance

La mise en correspondance consiste à mettre en correspondance les indices visuels du modèle avec ceux de l'image courante. Chaque caractéristique  $f^I$  est associé à son plus proche voisin du point de vue de l'apparence parmi les indices du modèle, noté  $f^M$ . Il est à rappeler que le plus proche voisin est déterminé grâce à la distance euclidienne sur notre descripteur. Soit  $E = e_1, e_2, \dots$  l'ensemble des appariements et  $e_k = i$  indiquant que  $f_k^i$  est apparié à  $f_M$ . Il a été proposé d'utiliser un seuil global sur la distance euclidienne pour éliminer les indices de fond. Cependant, cette méthode n'est pas efficace. Lowe a introduit une méthode de validation des appariements en utilisant le ratio entre la distance au plus proche voisin et la distance au second plus proche voisin. Ce ratio étant fixé à 0.8.

### Recherche de la vue clé la plus proche

L'objectif est d'identifier la vue clé qui correspond le plus au point de l'image actuellement traitée et ensuite d'éliminer les appariements inconsistants avec cette vue clé L'outil que nous avons utilisé est la transformée de Hough généralisée sur l'espace discret des similitudes 2D (

## 2. INDEXATION LOCALE DES IMAGES

---

translation, rotation, échelle) et des vues-clé. Un appariement  $e_k$  permet d'avoir les paramètres d'une similitude pour chaque vue clé associée  $f_{e_k}^M$  et vote pour les entrées de l'accumulateur de Hough. L'espace de transformation est discrétisé, ce qui engendre les effets de bord. Afin de les éviter, chaque appariement vote pour les deux entrées les plus proches de chaque dimension de la similitude.

### Vérification géométrique

Nous appliquons une vérification géométrique de la consistance de l'appariement obtenu précédemment afin d'intégrer les informations de la nouvelle image. Nous cherchons une évaluation plus précise de la transformation  $T_j$  par résolutions au moindre carré d'un système linéaire au cas où nous avons suffisamment d'appariements d'indices visuels.

Un point du modèle  $(x, y)$  est lié à un point de l'image  $(u, v)$  par l'équation :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} s \cos(\theta) & -s \sin(\theta) \\ s \sin(\theta) & s \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (2.54)$$

$t_x$  et  $t_y$ , la translation de l'image,  $\theta$  la rotation de l'image,  $s$  l'échelle de l'image.

Avec l'extraction des paramètres de similitude, nous obtenons un système linéaire  $Ax = B$  à partir de l'ensemble des appariements. Chaque appariement donne deux équations comme indiqué :

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} m \\ n \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{pmatrix} \quad (2.55)$$

Tel que :  $m = s \cdot \cos(\theta)$  et  $n = s \cdot \sin(\theta)$

La résolution de ce système aux moindres carrés est donnée par :

$$x = [A^T A]^{-1} A^T b \quad (2.56)$$

### Mise à jour du modèle

La décision de mise à jour dépend de la qualité de la mise en correspondance et l'erreur moyenne de la transformation entre la vue et l'image. Lowe dénombre trois cas :

1. Si le nombre des appariements est insuffisant, il estime que le point de vue de l'image courante est nouveau. Donc, nous intégrons les indices visuels au modèle et les associons à une nouvelle vue-clé.
2. Si l'erreur moyenne  $\epsilon$  est supérieure au seuil  $T$ , alors nous considérons qu'il existe bien une correspondance entre la vue-clé choisie et l'image courante mais que la différence

## 2.4 Reconnaissance d'objets

de point de vue est trop importante pour pouvoir intégrer les nouvelles informations. Alors, une nouvelle vue clé est créée. Mais les indices visuels  $f_k^I$  apparus ne sont pas ajoutés au modèle, uniquement un lien entre l'indice  $f_{e_k}^M$  correspondant et la nouvelle vue-clé est instancié.

3. Si l'erreur moyenne  $\epsilon$  est acceptable, il fusionne les informations de l'image courante et les vues clé sélectionnées. Il projette chaque  $f_k^I$  dans le référentiel de la vue clé  $V_j$ . Dans le cas où  $f_k^I$  est apparu, la projection est utilisée pour la mise à jour du lien  $l_{e_{kj}}$ . Sinon,  $f_k^I$  est intégrée au modèle et liée à la vue clé. (liée à la vue cl)

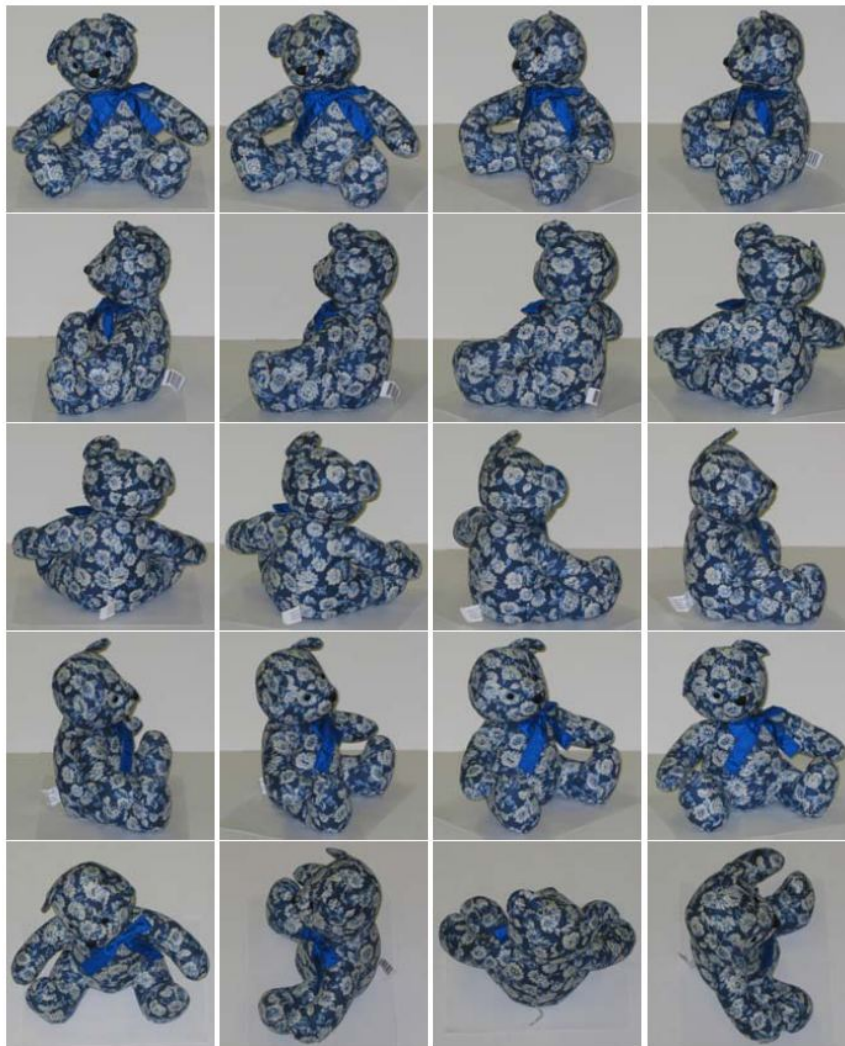


Figure 2.9: 20 images utilisées pour construire le modèle de Teddy. Il y a 16 images localisées dans un cercle équatorial. Cette configuration (avec une variation dans le nombre d'images d'entrée) est typique pour nos expériences de modélisation



## 2. INDEXATION LOCALE DES IMAGES

---

### 2.4.2 Résultats expérimentaux

### 2.4.3 Evaluation des performances

Nous présentons une évaluation standard de notre détection et description. Nous discutons après les résultats obtenus par la reconnaissance d'objets. Notre descripteur est évalué en utilisant une séquence d'image provenant du *Ponce Group Dataset* (Pon) et du *Amsterdam library of object images*.

### 2.4.4 Répétabilité

#### Définition :

Soient  $I_1$  et  $I_i$  deux images d'une même scène et  $M_1$  et  $M_i$  les matrices de projection correspondantes. La détection des points image  $p_1$  et  $p_i$  appartenant respectivement à  $I_1$  et  $I_i$  est répétable si et seulement si  $p_1 = M_1P$  et  $p_i = M_iP$ . Pour mesurer la répétabilité d'un détecteur, il faut établir une relation entre  $p_1$  et  $p_i$  à partir des images. Dans le cas général, il n'existe pas de telle relation. Toutefois, si  $I_1$  et  $I_i$  sont des images d'une scène plate, cette relation est définie par une homographie :  $p_i = H_{i1}p_1$  o  $H_{i1} = M_iM_1^{-1}$

Dans le cas d'une répétabilité parfaite entre  $I_1$  et  $I_i$ , on obtient les points détectés sur ces images  $p_1$  et  $p_i$  tels que:

$$H_{i1}p_1 = p_i$$

Dans des conditions réelles, un sous-ensemble uniquement de points est répété. En outre un point n'est souvent pas répétable de façon exacte, mais il est répétable dans un voisinage. Nous allons dans la suite introduire une mesure de répétabilité qui prend en compte ces deux faits.

L'algorithme qui calcule le ratio est présent dans ((MS04)) :

1. l'erreur dans la position du point relative est inférieure à 1.5 pixel :  $\|x_a - Hx_b\| < 1.5$ , où H est la matrice de l'homographie.
2. l'erreur dans la surface de l'image couverte par le voisinage du point est inférieure à 0.4. Dans le cas de l'invariance de l'échelle, l'erreur est :

$$e_s = \left| 1 - s^2 \frac{\min(\sigma_a^2, \sigma_b^2)}{\max(\sigma_a^2, \sigma_b^2)} \right| \quad (2.57)$$

où  $\sigma_a$  et  $\sigma_b$  sont les échelles sélectionnées et s est le facteur de l'échelle actuelle obtenu par l'homographie entre les images ( $s > 1$ )



## 2. INDEXATION LOCALE DES IMAGES

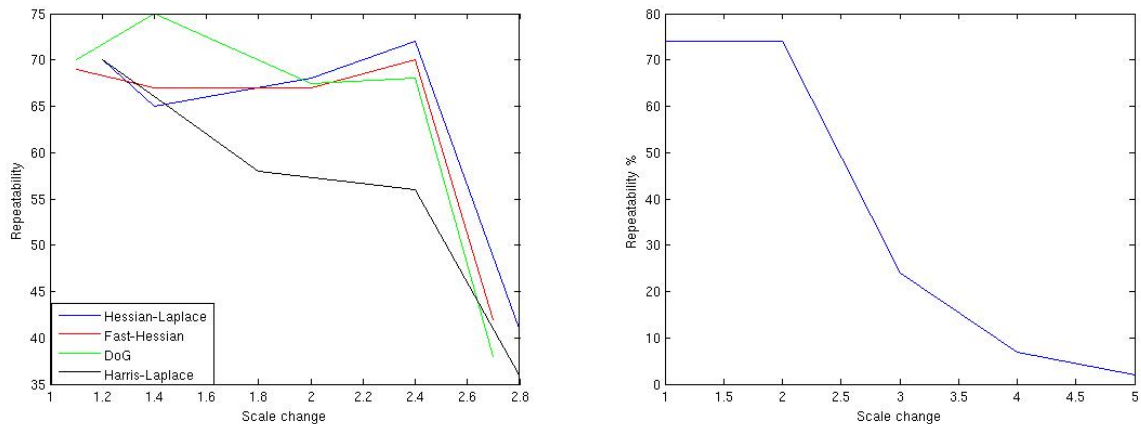


Figure 2.11: Première figure : Répétabilité du Fast Hessian, DoG, Harris-Laplace, Hessian-Laplace en fonction de l'échelle, Seconde Figure : répétabilité de notre détecteur en fonction de l'échelle

Nous estimons la stabilité de notre descripteur en effectuant des projections affines, des changements de contraste et de luminosité. Nous estimons la position de chaque point d'intérêt dans la première image en prédisant les paramètres de transformation dans la seconde image. Dans le tableau suivant, nous montrons la stabilité des points aux transformations d'images. Chaque ligne montre une transformation particulière de l'image. Alors que la colonne donne le pourcentage des descripteurs qui ont des correspondants dans l'image transformée.

La méthode de reconnaissance que nous utilisons est testée sur une base de données qui contient 160 images d'apprentissage et 51 images de tests sur un arrière plan uniforme. La scène de test contient entre 0 et 5 objets appartenant à l'ensemble d'apprentissage. Aussi le point de vue change d'une manière significative entre les images contenant le même objet.

Résultats :

Le tableau 3.1. montre les données d'apprentissage et du modèle de chaque objet :

### 2.4.5 Discussion :

Nous avons effectué l'apprentissage de divers objets pris sur un fond uniforme. Chacun des objets est représenté par  $N$  descripteurs  $f_i$ . Nous construisons une base de données de  $K$  objets ayant  $N$  caractéristiques chacun. Ensuite la reconnaissance est effectuée par l'extraction de  $N$  caractéristiques à partir de  $P$  images tests. En appliquant la transformée de Hough généralisée, nous arrivons à construire des clusters dans les images testes qui identifient les objets recherchés. Comme il a été mentionné avant, chaque caractéristique de l'image objet vote sur les caractéristiques similaires dans l'image test. D'autre part, nous obtenons une bonne répétabilité de notre descripteur, environ 75% pour une échelle de 1.5, 25% pour une

## 2.4 Reconnaissance d'objets

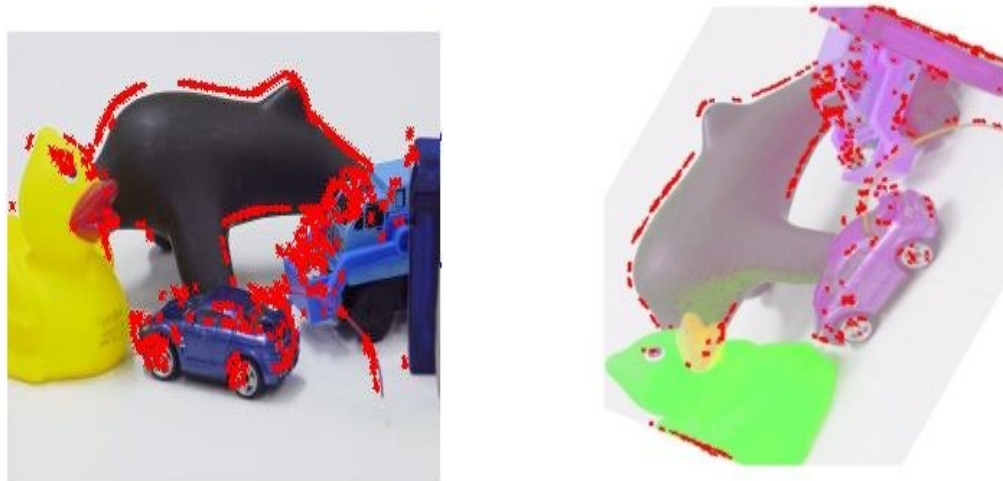


Figure 2.12: La deuxième image a été générée par le changement de la luminance, le contraste et en effectuant une rotation de la première image

Transformations d'image	Appariement%
Elever le contraste par 2	95.5%
Diminuer la luminance de 2	97%
Rotation de l'image par 2	94.5%
Changement de l'échelle par un facteur 0.7	97.25%

Table 2.1: Transformations appliquées à un échantillon d'images. Le tableau donne le pourcentage de clés appariées par appariement en utilisant la méthode K Nearest Neighbor (KNN)

## 2. INDEXATION LOCALE DES IMAGES

---

	Apple	Teddy	Box	shoes	Car
	Learning				
Images	29	20	16	16	28
Features	100	100	100	100	100
	Recognition				
Images	5	7	5	4	6
Features	100	100	100	100	100

Table 2.2: Apprentissage des modèles d'objets

échelle de 3 et 7% pour une échelle de 4; ce qui est comparable au détecteur de DoG (75% pour une échelle de 1.5). Notre descripteur est particulièrement applicable aux images en couleurs, décrites par le modèle de couleur gaussien, et aussi à la texture mesurée par les ondelettes de Gabor. Enfin, nous évaluons l'invariance en appliquant plusieurs transformations comme le changement d'échelle, de rotation et de luminosité (Élévation du contraste par 2 donne une invariance de 95.5% diminution de la luminosité de 2 donne une invariance de 97%).

### 2.5 Conclusion

Dans ce chapitre nous avons commencé par présenter l'état de l'art sur les détecteurs de caractéristiques invariantes dans des images. Nous avons développé un nouveau détecteur fondé sur la couleur et la texture; ce détecteur permet de trouver des points d'intérêt répétables dans l'image. Il assure l'invariance à la rotation et à l'échelle. Notre évaluation du caractère discriminant de ce descripteur, se fonde sur le développement d'un système de reconnaissance d'objets. Notre système de reconnaissance commence par apprendre les différents points de vue d'un objet. Ensuite il regroupe les points d'intérêt en clusters à travers la transformée de Hough généralisée qui cherche les points avec des paramètres similaires. Notre détecteur a une répétabilité comparable à celle du détecteur de Harris.

# Localisation des robots mobiles à partir de radio-étiquettes

## 3.1 Introduction

Dans ce chapitre, nous allons présenter nos méthodes de localisation par RFID et par amers visuels ponctuels. Nous proposons d’abord une approche déterministe fondée sur les RFIDs; puis nous adoptons une autre méthode stochastique qui utilise le filtrage particulaire (DFBT99) dont le principe a été rappelé en annexe 5.6. Nous présentons ensuite des méthodes d’optimisation basées sur les tags non observés et sur une approche de localisation active qui s’inspire des techniques actives de SLAM.

De nouveaux domaines s’intéressent au développement de la robotique comme l’assistance des personnes handicapées et l’intervention dans des zones dangereuses. Aussi, de nouvelles perspectives visent à introduire le robot dans les grandes surfaces comme les supermarchés. Notre objectif est d’assister un client à effectuer des achats dans un magasin. Nous présentons en figure 3.1 les démonstrateurs utilisés dans notre travail.

Ces robots sont équipés de différentes sortes de capteurs pour détecter, suivre et identifier une personne. Comme illustré en figure 3.2, un supermarché est un environnement très structuré où existent des couloirs qui séparent les étagères. L’espace libre dans le magasin peut changer très souvent par ordre du gestionnaire du magasin. Par conséquent, les chariots doivent utiliser leurs capteurs pour se localiser, mais ils devront aussi mettre à jour la carte pour tenir compte des changements fréquents de ce type d’environnement. A terme, les trolleys feront cette mise à jour de manière coordonnée, en exploitant aussi une communication robot-robot.

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---



Figure 3.1: A gauche, le trolley robotisé. A droite, le robot RACKHAM

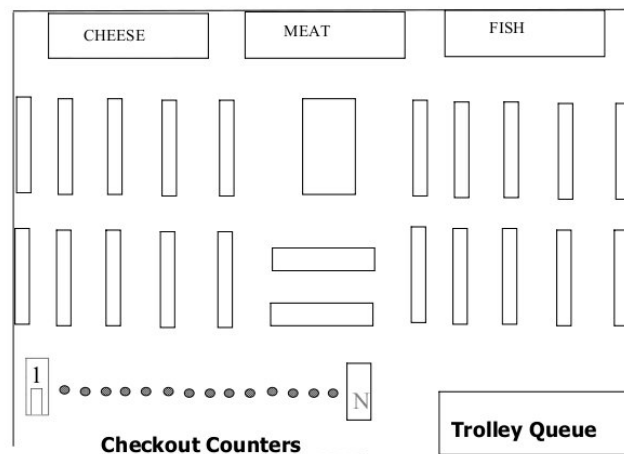


Figure 7.1: The supermarket metrical model, provided by the supermarket manager

Figure 3.2: Schéma du magasin

### 3.2 Localisation

Pour naviguer, le robot doit répondre aux trois questions fondamentales pour la tâche de navigation : Où suis-je? Où sont les autres lieux par rapport à moi ? et Comment puis-je atteindre mon but?

Le processus complet qui permet à un robot de mémoriser son environnement, puis de s'y déplacer pour rejoindre un but, peut être découpé en trois phases: la cartographie, la localisation et la planification.

1. La cartographie est la phase qui permet la construction d'une carte reflétant la structure spatiale de l'environnement à partir des différentes informations recueillies par le robot.
2. A partir d'une telle carte la localisation permet alors de déterminer la position du robot dans la carte qui correspond à sa position dans son environnement réel.
3. La planification, enfin, est la phase qui permet, connaissant la carte de l'environnement et la position actuelle du robot, de prévoir les mouvements à effectuer afin de rejoindre un but fixé dans cet environnement.

Ces trois phases sont évidemment fortement interdépendantes. L'ordre dans lequel elles sont citées fait directement apparaître le fait que la seconde phase dépend de la première. En effet, estimer sa position au sein d'une carte de l'environnement suppose implicitement que cette carte existe et qu'elle contient la position courante du robot. De même, la troisième phase dépend des deux premières, car la planification suppose que l'on connaisse sa position et que la carte de l'environnement représente une portion de l'environnement contenant au moins un chemin reliant cette position au but qui doit être atteint. Mais la relation entre les deux premières phases est plus subtile qu'une simple relation d'antériorité : c'est le même problème que celui de l'oeuf et la poule. Chacun des deux éléments peut, en effet, être considéré comme préalable à l'autre mais dépend aussi de l'autre pour sa réalisation. Dans le cas de la cartographie et de la localisation, nous avons déjà vu que la localisation repose sur une phase préalable de cartographie. Mais pour construire une carte, il est nécessaire de savoir où ajouter, dans la carte partielle déjà existante, toute nouvelle information recueillie par le robot. Cela requiert donc une phase préalable de localisation au sein de la carte partielle déjà existante.

Nous décrivons plusieurs méthodes de self localisation basée sur des RFID. Notre but est de guider un utilisateur jusqu'à une étagère où se trouve le produit qu'il souhaite acheter. La position du robot est exprimée par le vecteur  $(area, x, y, \theta)$ . Ceci permet de faire une



### **3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES**

---

localisation topologique dans laquelle le robot peut se déplacer d'une zone à une autre et métrique pour savoir la position du robot par rapport à une référence de base.

La localisation est l'un des problèmes fondamentaux de la robotique, les robots mobiles doivent en effet se localiser dans un environnement opérationnel pour effectuer leurs tâches.

Il y a trois approches principales de localisation : L'approche métrique, topologique et hybride.

Dans l'approche métrique, la position est estimée d'une manière aussi précise que possible par rapport au systèmes de coordonnées de la carte.

Dans l'approche topologique, l'environnement est représenté par des points connectés. Cela signifie que le robot mémorise plusieurs places distinctes et tente de déterminer le noeud pour les places qui correspondent à la position du robot. Alors que la méthode hybride combine les deux. Les deux premières méthodes font l'appariement en utilisant l'appariement de la carte et la détection des landmarks Le problème du frottement avec le sol reste un problème pour les robots de type Marsokhod. En fait, seul les glissements latéraux permettent au robot de faire des rotations. Ceci est affecté par la nature du sol.

#### **3.2.1 RFID**

Les tags RFID (aussi appelés " étiquettes RFID ") sont de petites étiquettes qui sont collés sur les objets. Les plus grands ressemblent à de petits autocollants carrés sur lesquels se trouve une puce électronique, les plus petits ne sont pas plus grands que la mine d'un crayon. La technologie RFID est assez récente. Elle s'est développée rapidement depuis 10 ans, autant à cause des attraits technologiques qu'elle suscitait que du prix qui baissait. Le tag est composé d'une antenne lui permettant de capter l'énergie que lui envoie le lecteur RFID, associée à une puce électronique, qui lui permet de répondre aux requêtes radio émises depuis l'émetteur-récepteur.

Sur l'image précédente, nous observons bien l'antenne du tag autour de la puce. Il existe plusieurs types de tags RFID, les actifs et les passifs. Même si le principe est similaire pour les deux types, certaines différences fondamentales existent.

##### **Les tags passifs**

Les tags passifs ne possèdent pas de batteries, cela implique qu'ils sont en veille la plupart du temps. Lorsque le tag reçoit une onde provenant d'un lecteur, il emmagasine grâce à l'antenne, l'énergie nécessaire au tag pour renvoyer l'information demandée. A cause du manque de batterie, la portée des tags passifs n'est pas très élevée.

##### **Les tags actifs**

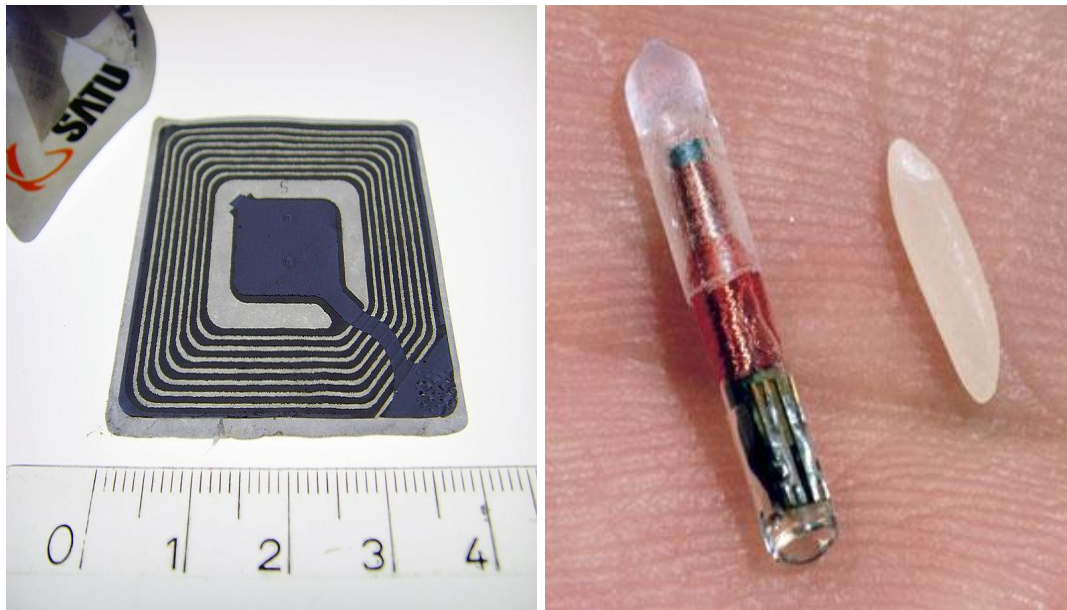


Figure 3.3: (A gauche) antenne RFID (A droite) tags RFID

Sont équipés d'une batterie qui peut être utilisée comme une source de puissance pour le circuit des tags et des antennes.

Notre démonstration dans le cadre du projet COMMROB est basée sur un kit comprenant un lecteur, 8 antennes, 10 tags. Nous avons équipé notre robot Rackam par un lecteur RFID supportant seulement 4 antennes. Nous avons donc été amenés à faire un multiplexeur pour pouvoir supporter les 8 antennes que nous possédions. Il était plus simple d'un point de vue implémentation et surtout plus économique de faire de cette manière plutôt que de racheter un autre lecteur.

Le but de notre carte était de permettre au lecteur d'être connecté la moitié du temps aux antennes de droite, et l'autre moitié, à celle de gauche. Nous avons choisi pour commuter les antennes, d'utiliser une broche du connecteur RS-232 qui n'est pas utilisée ( RTS ). Nous avons donc dû réaliser la commande du boîtier qui sera plus tard relié aux 4 switches.

Le but premier de l'utilisation des tags RFID est de permettre la détection précise de la position de l'utilisateur du robot par rapport au robot. En effet, il est essentiel que le robot interagisse avec son utilisateur et non pas avec les autres personnes à proximité. Pour cela, l'utilisateur se voit pourvu d'un tag à identifiant unique. Le robot pourra donc obtenir des informations sur la position du porteur de tag dans son environnement. Ces informations ne seront pas précises mais donneront un ordre d'idée de la position et de l'angle du porteur du tag par rapport au robot.

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

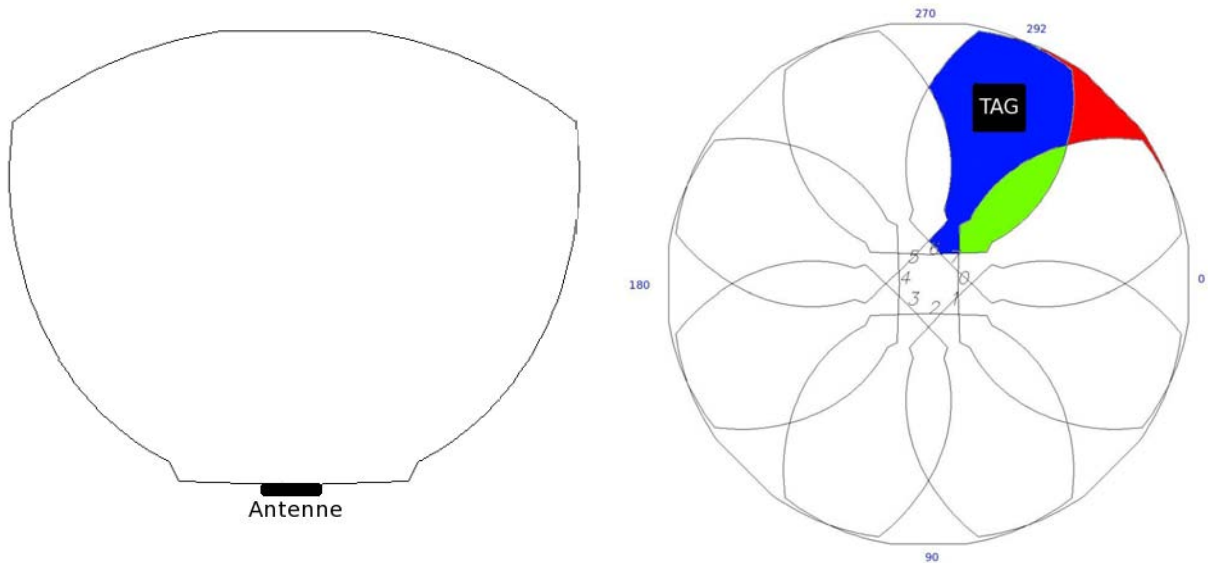


Figure 3.4: Champ de réception des 8 antennes montées sur un robot circulaire

Tout d'abord il a fallu caractériser la zone de détection d'une antenne, il a ensuite été possible de regrouper ces zones pour obtenir les différentes configurations possibles lors de la détection d'un tag. La carte suivante a donc été créée.

On voit sur l'illustration dans la 3.4 (droite) que 4 cas sont possibles :

1. Aucune information supplémentaire n'est disponible. Le tag peut être à une distance de plus de 4.5m ou être occulté par quelque chose. Dans le cas qui nous intéresse, un tag non présent dans la liste des tags détectés, n'est pas pris en compte. Seule l'information " tags détectés " est considérée comme utile.
2. Le tag est détecté par une, deux ou trois antenne.

On voit bien sur le schéma que même si les informations obtenues ne sont pas précises, on ne peut pas savoir à quelle distance est le tag ou dans quelle direction, et que l'on peut tout de même donner un ordre de grandeur. Dans le cas où le tag est détecté par une antenne (zone rouge), la distance au tag est comprise entre 3.2m et 4.5m. De même l'angle est compris entre -23 et 23 degrés autour de la position moyenne de l'angle. Cette position moyenne est obtenue en fonction du nombre d'antenne ayant détecté le tag et du numéro de celle ci. Par exemple, dans l'illustration (6.2 droite), le tag est détecté par deux antennes : la 6 et la 7. On peut calculer aisément que le milieu de la zone a pour angle 292 degrés par rapport au robot. On sait aussi que dans cette zone, l'angle peut

varier de -25 à +25 degrés par rapport à la position moyenne. De même, la distance du tag au robot peut varier de 0.5 à 4.5m. Ces informations même imprécises peuvent être utiles lors de la localisation de l'utilisateur.

Grâce à ces données, il est possible de déterminer une zone probable où se trouve l'utilisateur, cette zone étant plus ou moins restreinte selon le nombre d'antennes détectant le tag. On pourra alors utiliser différents capteurs pour affiner la localisation de l'utilisateur.

### 3.2.2 Etat de l'art sur la navigation par RFID

Actuellement, parmi les applications en navigation robotique, on trouve l'estimation de la trajectoire et de l'odométrie à l'aide d'RFID à prix réduit. L'estimation de la trajectoire est un pré-requis pour le mapping via RFID sans une position de référence. En utilisant le filtrage particulaire (présenté dans l'annexe), on se base sur un modèle non paramétrique des relations spatiales entre les mesures RFID. L'avantage de cette méthode est qu'elle évite la nature bruitée des mesures aussi bien que l'absence des informations de la distance et le "bearing". Les positions du transpondeur ne sont pas cartographiées simultanément. En plus, une reconstruction consistante de la trajectoire du robot implique l'intervention des entrées des capteurs et la fermeture de boucle. ((MTKW02)) ont travaillé sur la génération de cartes avec des télémètres laser. Hanel et al ((HBF+05)) ont utilisé le Fast SLAM fondé sur le laser pour calculer la trajectoire de leur robot et la position des tags UHF-RFID passifs. Après avoir calculé la carte des transpondeurs, le robot est finalement capable de se localiser avec les RFID et l'odomètre. Dans notre contribution, nous nous sommes seulement placés dans ce contexte et nous avons considéré que la carte était apprise avant. ((KCS07)) ont effectué une correction de la trajectoire en utilisant du graphSLAM avec des transpondeurs actifs. En ce qui concerne les tags actifs, Kantor et al ((KSP+03)) ont exploité le filtrage de Kalman étendu pour la localisation, la cartographie et le SLAM. Ils exploitent la puissance de signal mesurée entre les transpondeurs. Cette dernière n'est pas standardisée dans les systèmes RFID passifs. Par ailleurs, une autre approche utilisée entre autre par Vorst ((VSYZ08)) utilise les empreintes (fingerprints) pour modéliser les observations des tags. L'idée clé des empreintes est la comparaison du tag courant détecté et une base de données des mesures obtenues durant l'apprentissage.

Le travail de Hanel et al ((HBF+05)) est considéré parmi les premiers travaux de localisations de robots par le biais des RFIDs. Un modèle probabiliste de capteurs des lecteurs RFID est utilisé. Ceci consiste à associer la probabilité de détection de tags avec la position du tag par rapport aux antennes. Ce modèle est utilisé pour avoir la position des tags RFID en

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

utilisant une carte pré-calculée, apprise durant une phase de SLAM par laser. En effet, la position du tag est représentée par un nombre de particules. Les poids de ces particules sont mis à jour après chaque détection de tag. Quant à la localisation du robot, un autre ensemble de particules est utilisé. Lu a développé un système pour les tags UHF-RFID passifs qui exploitent la direction des lecteurs RFID. La probabilité des positions des objets et tags associés est calculée en variant la position du robot en fonction du temps. Une autre approche de ((YTH<sup>+</sup>04)) utilise les SVM pour localiser les robots. Cela consiste à générer un vecteur de points à partir de la puissance du signal qui provient des tags RFIDs actifs.

Tsukiyama et al ((Tsu03)) ont développé un mécanisme simple de navigation utilisant la vision pour la détection des espaces vides et les tags RFID en tant que labels sur une carte topologique dans un environnement intérieur. Récemment, ((ZLH07)) ont proposé une méthode de localisation en milieu intérieur orientée vision et ont exploité une version modifiée de tags RFID actifs comme des amers. Ces tags ont été équipés avec des LED pour pouvoir être localisés.

#### 3.3 Localisation déterministe par RFID

Nous commençons par le calcul de la fonction d'observation. Ensuite, nous effectuons la prédiction en utilisant un modèle de mouvement et nous filtrons pour estimer la position nouvelle du robot.

**Calcul de la fonction de l'observation :** En nous basant sur le modèle de perception des tags exprimés, par les inéquations suivantes

$$\begin{aligned}(X_t + X_r)^2 + (Y_t + Y_r)^2 &< R_2 \\ Y_t - [\tan(\theta_a - \alpha)(X_t - X_a) + Y_a] &< 0; \\ Y_t - [\tan(\theta_a + \alpha)(X_t - X_a) + Y_a] &< 0;\end{aligned}$$

Nous calculons pour chaque vraie position, les tags reçus par l'antenne. Après avoir parcouru toutes les positions du robot, nous déterminons la fonction d'observation  $h$ .

**Prédiction** La position prédite est entachée de bruit blanc gaussien. Cette position est calculée en utilisant le modèle de l'odométrie du robot.

$$x_{t+1} = x_t + u$$

. où  $u$  est le déplacement.

#### Estimation

L'estimation  $X_e$  est présentée avec l'ellipse d'incertitude dans laquelle la vraie position doit être avec une probabilité de 0.95%. On trace autour de la position prédite une ellipse telle que la variance est calculée par :

$$P_3^* = \frac{dF}{dX} P_2^e \frac{dF^t}{dX} + \frac{dF}{du} Q \frac{dF^t}{du}$$

On discrétise cette ellipse. Autour de chaque particule on vérifie si elle reçoit les mêmes tags que la vraie position. Si c'est le cas, on la garde ; sinon on la rejette. Ensuite on calcule la moyenne arithmétique de toutes ces particules. Cette moyenne représente la position estimée du robot.

La méthode développée présente un inconvénient. En effet, elle ne prend pas en compte les antennes du robot qui n'ont pas d'observation. Nous utilisons ces informations pour avoir une localisation plus précise. Les étapes de notre méthode des non-observations sont :

- calculer l'observation du robot dans l'étape courante basée sur le modèle des antennes ;
- trouver les particules autour de la position prédite avec la covariance  $P_y$  ( Calculée dans l'étape de mise à jour de Kalman) qui reçoivent les mêmes tags RFID que l'observation ;
- rejeter les particules qui reçoivent d'autres tags.

Nous effectuons des mesures statistiques pour montrer l'amélioration des performances avec les non observations. Pour cela, nous faisons déplacer le robot dans son environnement avec différentes valeurs de bruit. Pour chaque cycle du robot, on calcule la variance de la mesure  $x_{Est} - x_{true}$  pour les deux cas, avec et sans amélioration. Les résultats sont présents sur la figure 3.8.

#### 3.3.1 Localisation par la méthode Range and Bearing

Le robot se déplace sur un plan et perçoit la direction d'amers ponctuels. L'état du robot est représenté par sa position et son orientation.  $X_t = [x_t, y_t, \theta_t]^T$

Le déplacement du robot entre les instants t et t+1 est mesuré grâce à l'odométrie donnée dans le repère du robot à l'instant t.

$$X_t = [x_u, y_u, \theta_u]^T$$

Le robot se déplace selon le modèle d'évolution suivant :

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

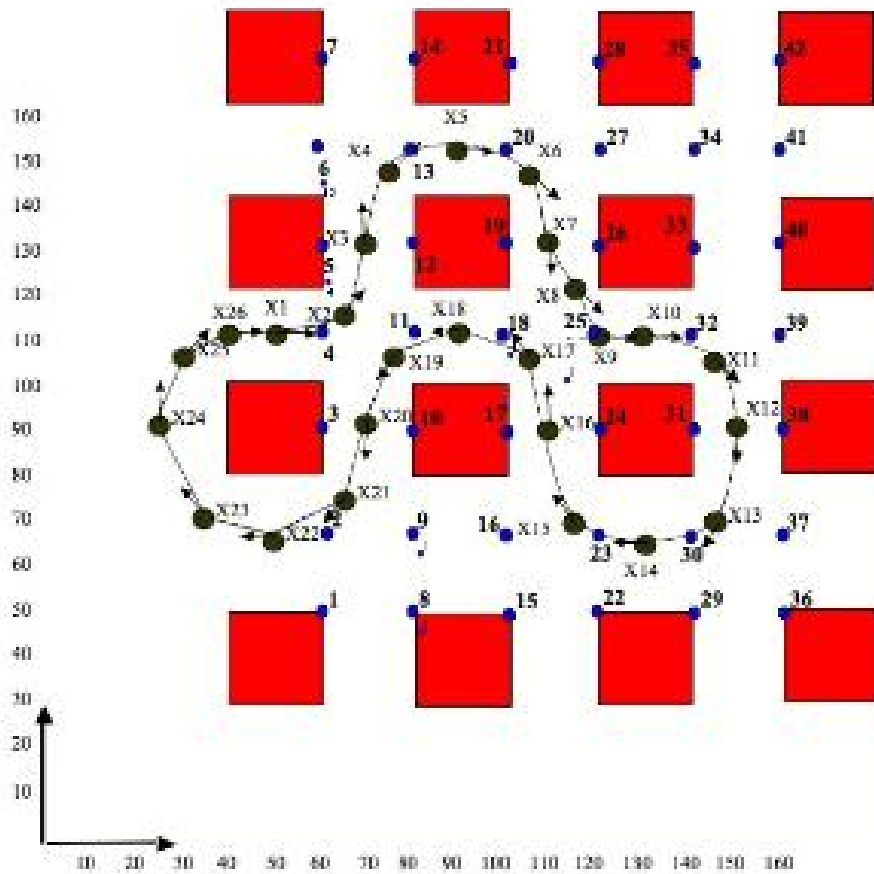


Figure 3.5: Un environnement simulé avec les tags en bleu et la trajectoire  $X_i$

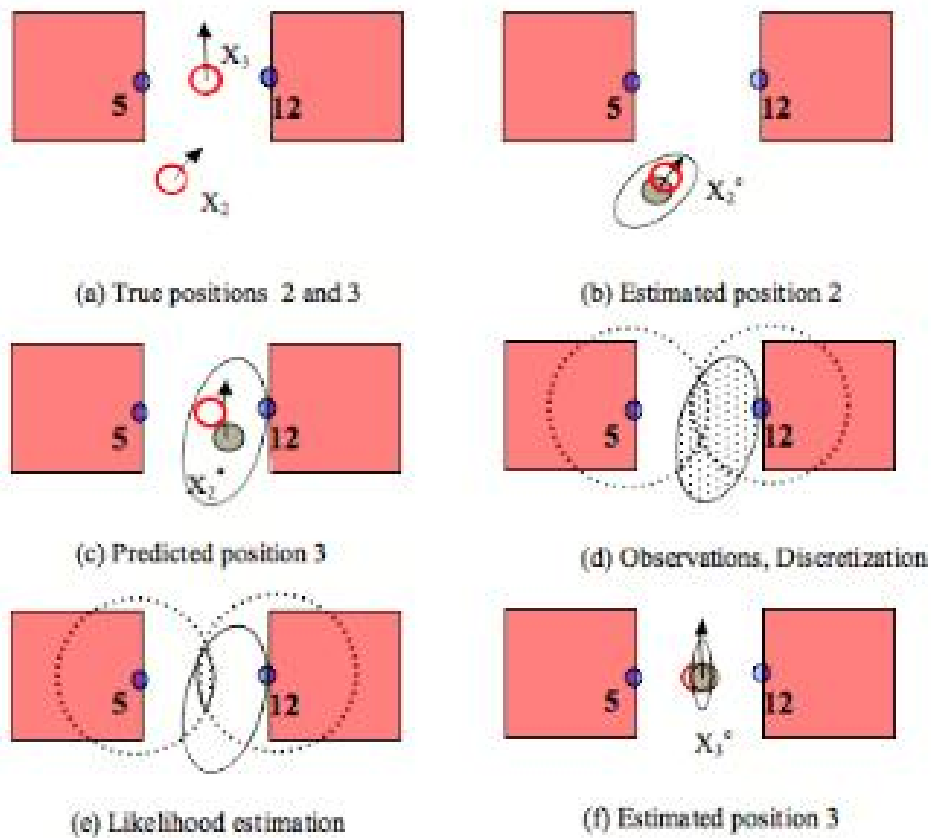


Figure 3.6: Prédiction, Observation, Estimation



### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

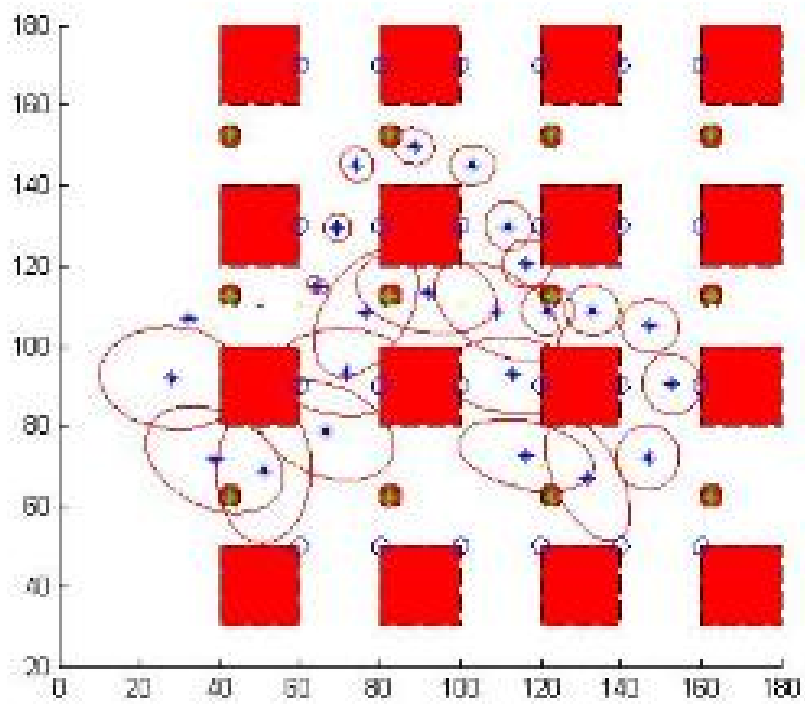


Figure 3.7: Les positions prédites à partir du modèle du robot ( odometrie )

### 3.3 Localisation déterministe par RFID

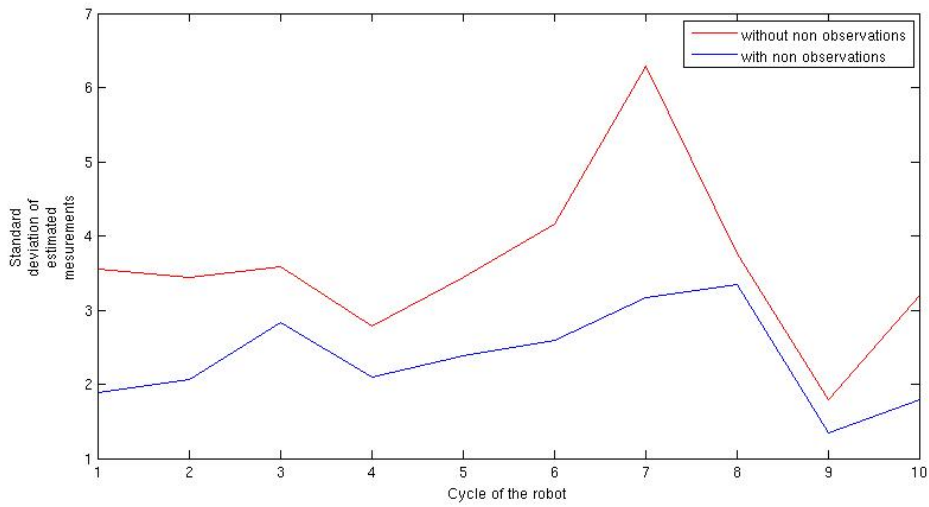


Figure 3.8: La variance de  $x_{Est}-x_{True}$  pour 10 cycles de robot, L'application des non-observations réduit l'erreur. La ligne rouge est pour les non observation, la ligne bleue est pour les non-observations

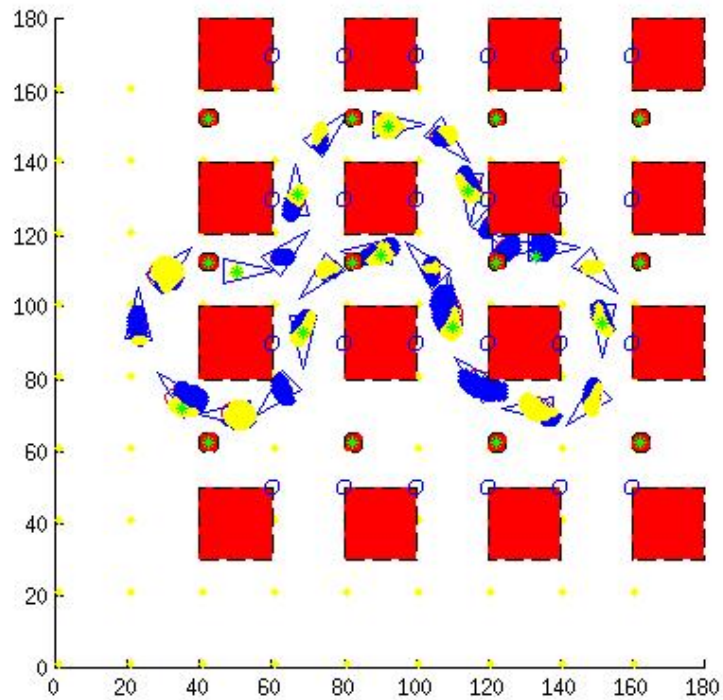


Figure 3.9: L'évolution de la distribution des particules( Avec la prise en compte des non-observations, nous gardons uniquement les particules jaunes)

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

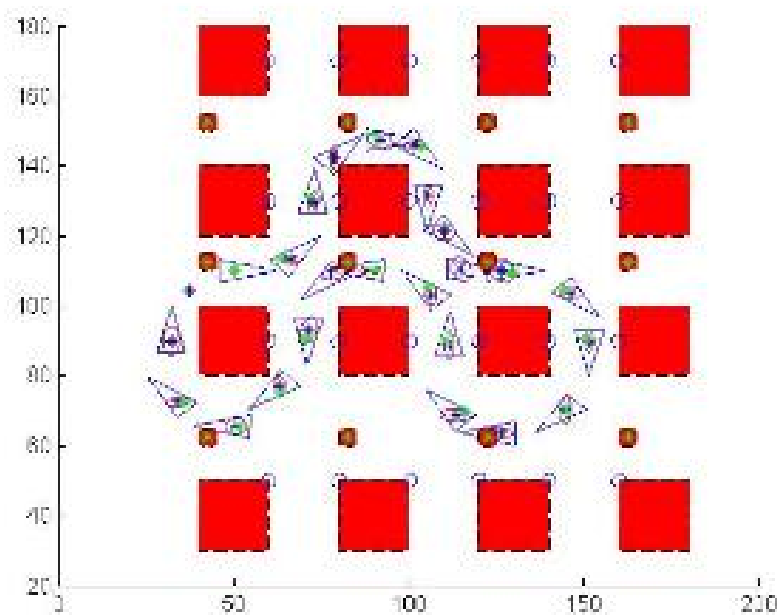


Figure 3.10: Localisation du robot à différentes positions du robot, avec le calcul de la variance de  $x, y, \theta$

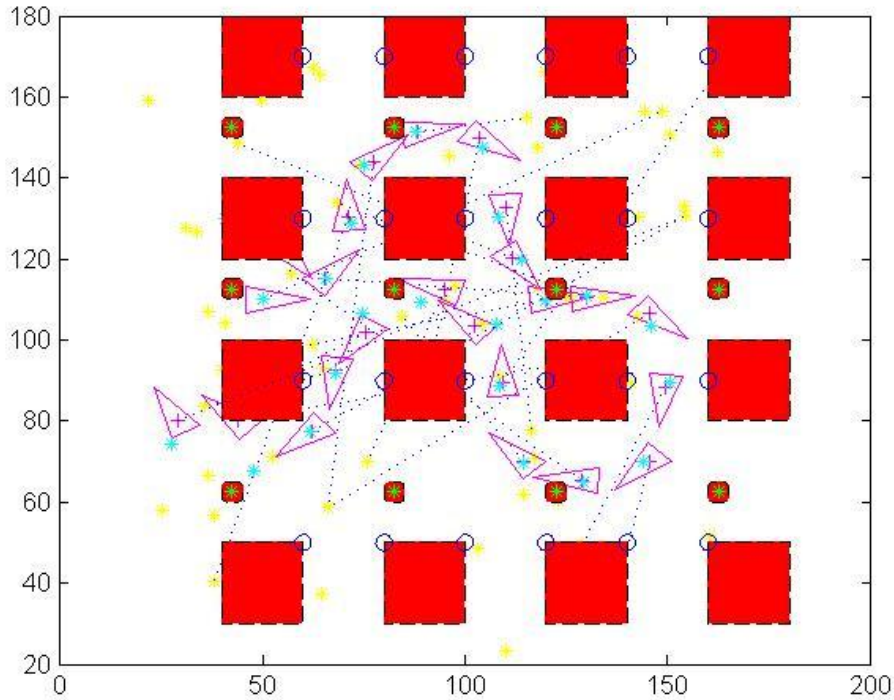


Figure 3.11: Recalage de la position du robot par amers visuels ponctuels

$$X_{t+1} = F(X_t, U_t) = \begin{bmatrix} x_t + x_u \cos(\theta_t) - y_t \sin(\theta_t) \\ y_t + x_u \sin(\theta_t) + y_t \cos(\theta_t) \\ \theta_t + \theta_u \end{bmatrix}$$

Avec un bruit gaussien donné par une matrice de covariance  $Q$ . Les perceptions fournissent des mesures de la distance et de la direction d'un amer  $k$  supposé parfaitement identifiable :

$$Z_t = [r_t^k, phi_t^k]^T$$

Le modèle d'observation correspondant est donc :

$$Z_t = h^t(X_t) = \begin{bmatrix} ((delta)) \\ \arctan(\frac{delta(2)}{delta(1)}) - \theta_t \end{bmatrix}$$

où  $x_k$  et  $y_k$  sont les coordonnées (connues) de l'amer dans le repère global. Ce modèle est entaché d'un bruit gaussien de matrice de covariance  $P_Y$ . Pour avoir l'ensemble des positions du robot, nous appliquons le filtrage de Kalman étendu. Nous présentons dans ce qui suit, les résultats de notre simulation :

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

#### 3.3.2 Localisation par des amers visuels ponctuels

Nous plaçons dans notre environnement des amers visuels qui sont en théorie ponctuels. Notre robot est équipé d'une caméra monoculaire. Il utilise la vision pour percevoir son environnement. La caméra permet en fait au robot de localiser les objets sur le terrain, en d'autres termes d'évaluer la position relative au robot. Aussi, elle permet d'évaluer sa position et son orientation sur le terrain. Nous utilisons, le modèle sténopé pour avoir la localisation du robot. Après avoir obtenu les coordonnées de l'image dans le plan image, on applique un filtrage de Kalman étendu((ATSB04)). L'innovation est donnée par :  $Innov = z - z_{pred}$ . Avec,

$$Z_t = h^t(X_t) = \begin{bmatrix} \sqrt{((delta)^2)} \\ \arctan(\frac{delta(2)}{delta(1)}) - \theta_t \end{bmatrix}$$

Et,

$$delta(1) = u - x_{veh}$$

$$delta(2) = v - y_{veh}$$

Nous considérons  $x_{veh}, y_{veh}$  et  $\theta_{veh}$  les coordonnées du robot

$$[x_{veh} y_{veh}] = [x_{pred} y_{pred}]$$

pour  $z_{pred}$

et

$$[x_{veh} y_{veh}] = [x_{true} y_{true}]$$

pour  $z$

D'où :  $x_{Est}$ , la position estimée est calculée par :

$$x_{Est} = x_{Pred} + Innov.K$$

K est le coefficient de Kalman.

#### 3.3.3 Localisation stochastique par filtrage particulière

Nous fixons pour notre simulation le nombre de particule à 300, le nombre de tirages dans la distribution de probabilité de l'odomètre pour chaque échantillon fixé à 3, le nombre de particules aléatoires injectées est fixé à 30. Puis on fera une brève analyse de l'influence de

### 3.3 Localisation déterministe par RFID

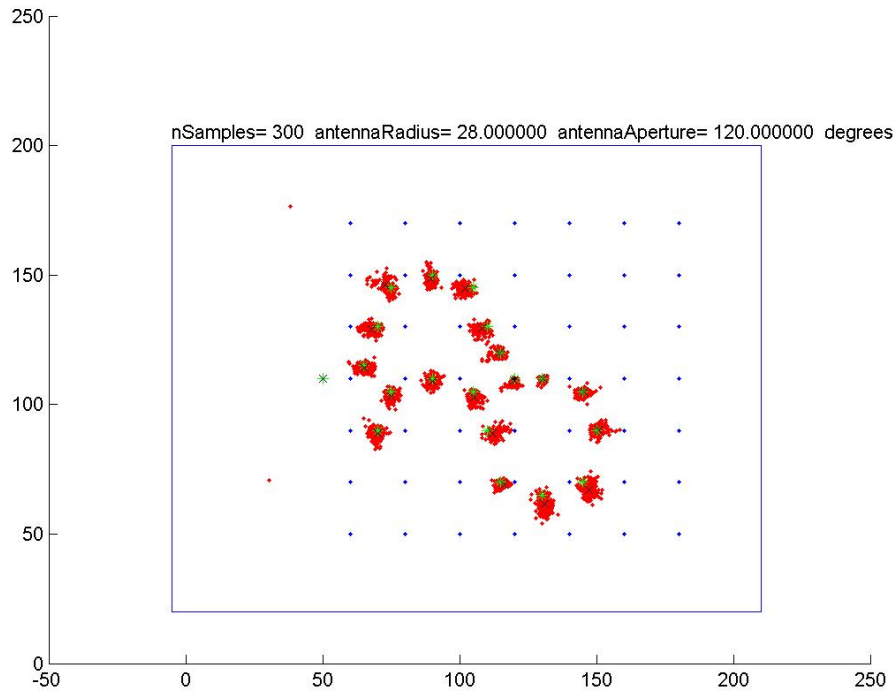


Figure 3.12: Simulation stochastique - évolution dans un milieu riche en tags

ces paramètres sur les résultats obtenus. Dans la figure 3.12, on présente une agrégation des premiers 20 déplacements du robot dans un milieu riche en tags. On a en rouge l'ensemble des échantillons, en noir la position estimée qu'ils donnent et en vert la position réelle. On remarque la faible étendue des taches rouges, qui correspond à une faible erreur sur l'estimation du positionnement

Du moment où on commence à ne plus recevoir des observations (on rentre dans la partie du milieu qui est pauvre en tags), on remarque que notre estimation commence à être de moins en moins bonne. Toutefois, cela ne pose pas un grand problème, car en regardant ce qui se passe une fois rentré entre les tags, on remarque que le robot regagne en précision.

On remarque aussi le fait que dans le cas où le robot commence à «se perdre», les particules aléatoires commencent à faire partie de l'état du système (Figure 3.11). Ceci va permettre au robot de se retrouver au cas où il se perdrait pour de vrai (c'est-à-dire, l'imprécision sur la position estimée deviendrait trop importante).

On simule un parcours de la trajectoire présentée dans la 3.14 cinq fois. On montre dans ce qui suit l'évolution des erreurs constatées .

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

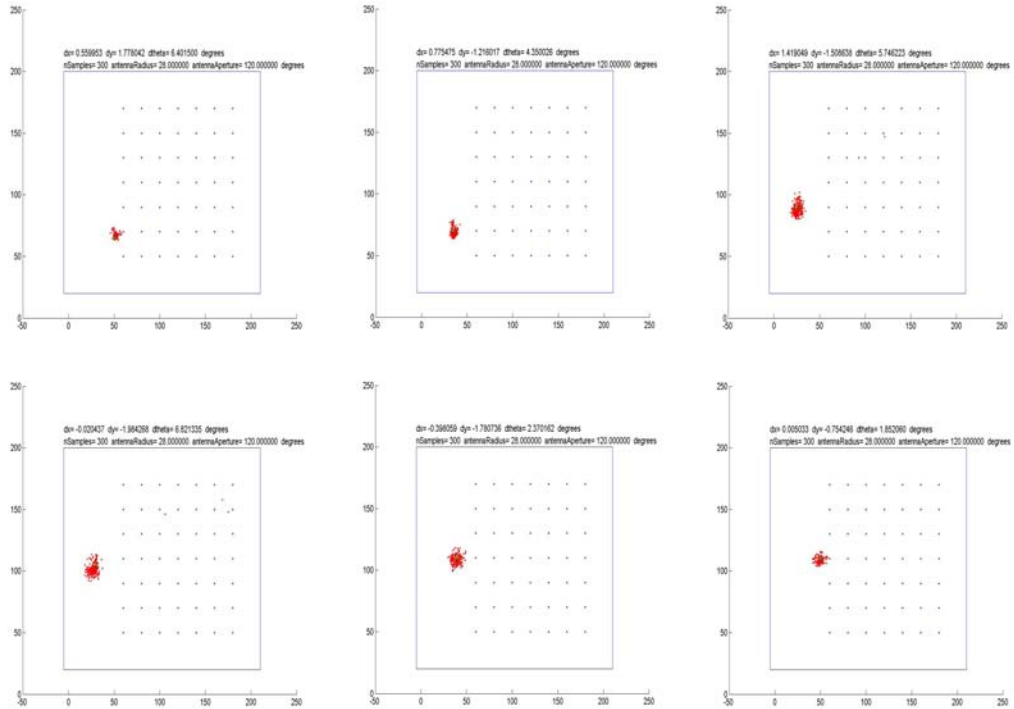


Figure 3.13: Simulation stochastique - évolution dans un milieu pauvre en tags

On voit effectivement que dans la plupart du temps l'erreur reste faible. Comme nous remarquons juste avant, elle peut croître dans les zones de l'environnement pauvres en tags (les intervalles verts sur le graphique) mais elle revient à des valeurs acceptables une fois que les observations recommencent. Pour cette simulation, les valeurs de l'erreur quadratique moyenne ont été  $\sigma_X = 1.991$ ,  $\sigma_Y = 1.539$ ,  $\sigma_\theta = 4.101$

Pour conclure cette section, on va regarder quelques comparaisons des performances de l'algorithme quand on varie ses paramètres principaux (le nombre de particules, le nombre de tirages dans la distribution odométrique, le nombre de particules aléatoires injectées).

La remarque principale que l'on peut tirer c'est qu'augmenter le nombre d'échantillons c'est gagner de la précision en sacrifiant du temps d'exécution. En effet, toutes les erreurs décroissent avec l'augmentation du nombre d'échantillons pendant que le temps d'exécution augmente linéairement .

Pour le nombre de tirages on observe le même comportement que pour le nombre d'échantillons. On perd toujours en temps d'exécution en gagnant de la précision. Bien qu'au niveau des performances il n'y a pas une grande différence, on note que les besoins en termes de mémoire d'une approche dans laquelle on augmente le nombre de tirages sont plus importantes que

### 3.3 Localisation déterministe par RFID

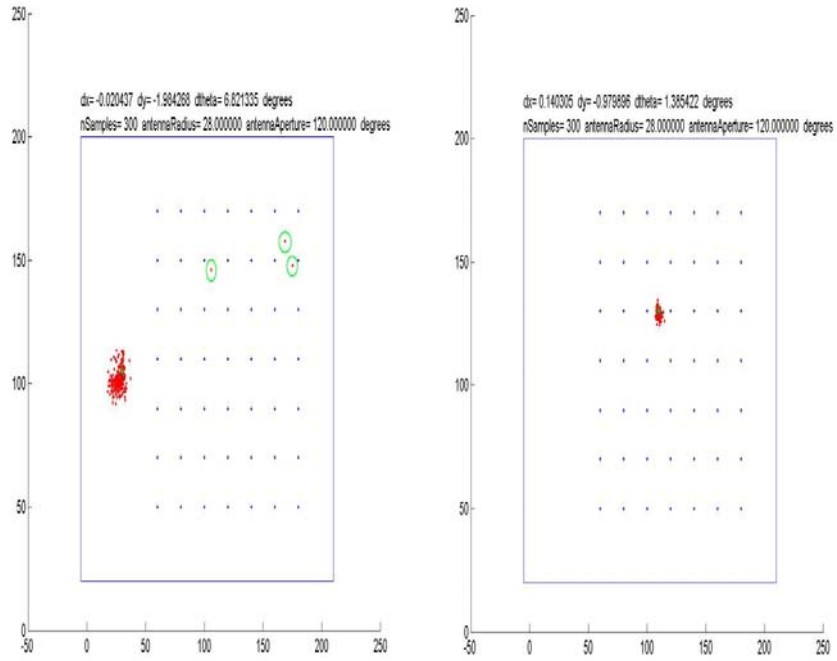


Figure 3.14: Simulation stochastique - évolution dans un milieu riche en tags

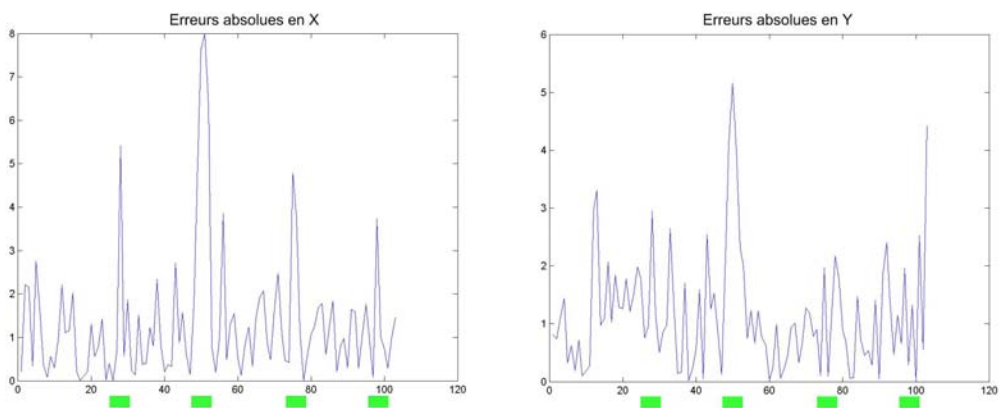


Figure 3.15: La variation de l'erreur en X et en Y



### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

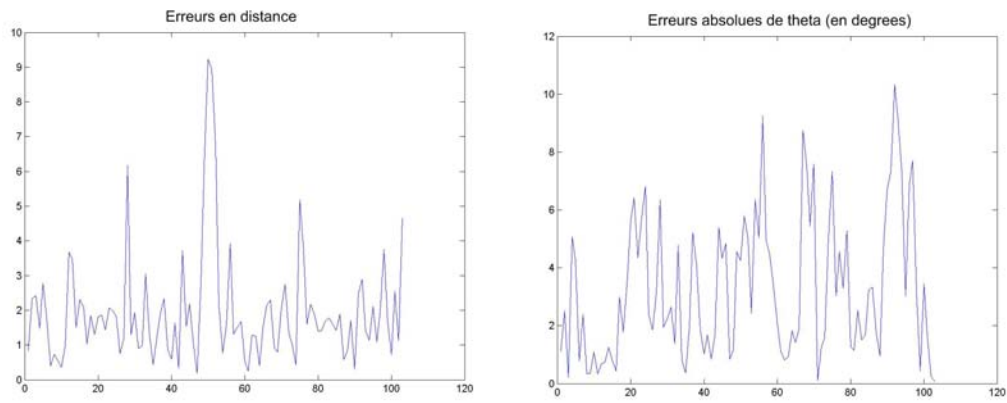


Figure 3.16: La variation de l'erreur en distance et en  $\theta$

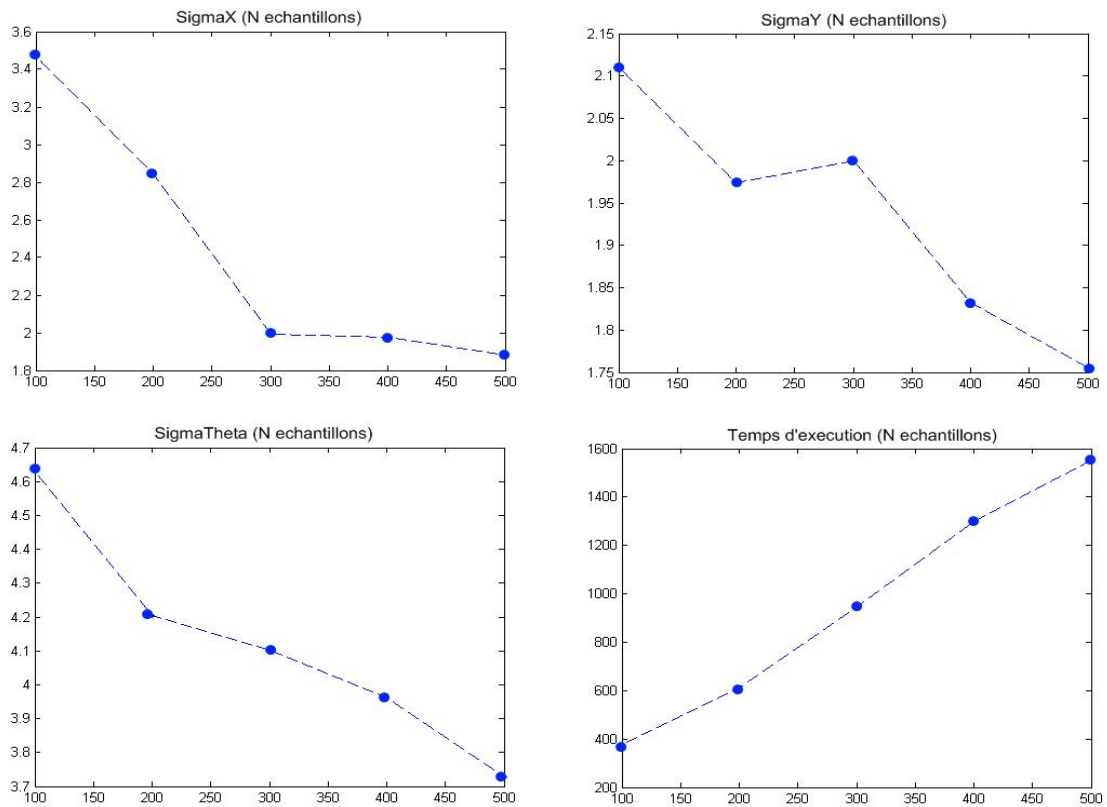


Figure 3.17: Comparaison des performances en variant le nombre d'échantillons

### 3.3 Localisation déterministe par RFID

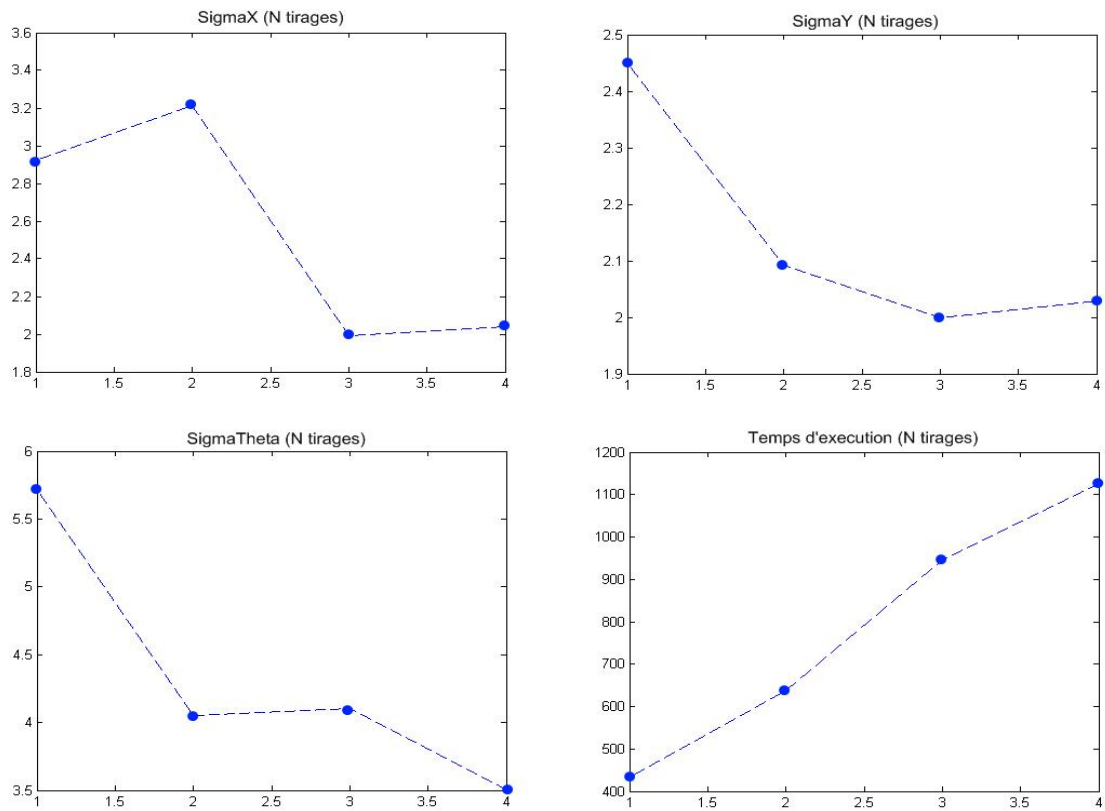


Figure 3.18: Comparaison des performances en variant le nombre de tirages

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

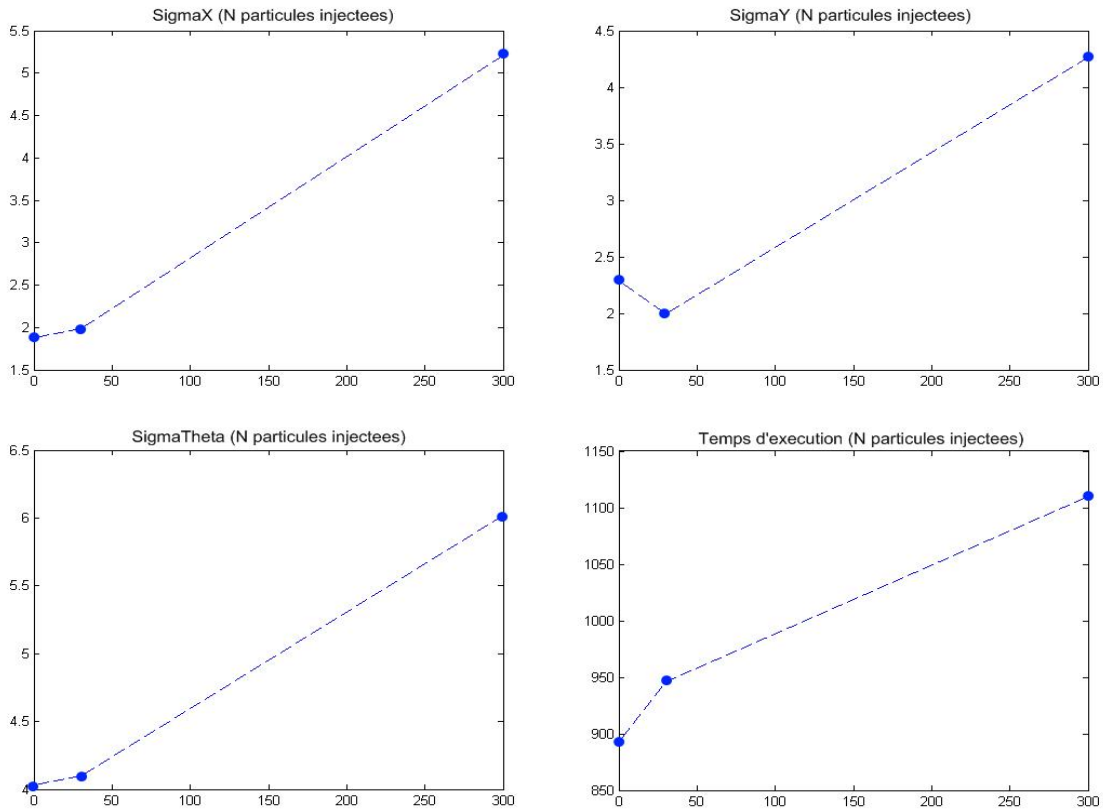


Figure 3.19: Comparaison des performances en variant le nombre de particules injectées

dans le cas contraire.

Dans le cas des particules aléatoires injectées, ce n'est pas le temps de calcul qui nous gêne, mais surtout le fait que la précision baisse. La conclusion c'est qu'un nombre pas trop grand (10% du nombre d'échantillons environ) n'entraîne pas de grosses pertes de précision et permet quand même au robot de se repositionner.

#### 3.4 Amélioration du recalage par localisation active

Afin de maximiser les performances de localisation, nous utilisons une stratégie de contrôle intelligent nommée localisation active (Bry09). Elle consiste à compter les actions de contrôle dans un processus d'estimation. On utilise pour cela un contrôle d'information. En effet, les mesures d'information quantifient l'incertitude dans une représentation probabiliste.

La métrique d'information est définie comme une fonction de distribution de probabilité (FW09).

### 3.4 Amélioration du recalage par localisation active

---

$$I[x] = f[p(x)]$$

où  $p(x)$  représente l'estimée de la position du robot  $[xy\theta]$ , qui est considérée comme une gaussienne avec la moyenne  $\hat{x}$  et la covariance  $P$ . La métrique d'information est :  $h(x) = \frac{1}{2} \log[(2\pi e)^n |P|]$ . Le gain d'information étant défini comme la différence dans l'information de notre estimation avant et après une action particulière. Quand le véhicule se déplace, il suit la procédure suivante :

1. calculer une trajectoire qui maximise  $I[x,a]$  (  $a$  étant l'action)
2. proposer plusieurs trajectoires
3. estimer les observations selon chaque trajectoire
4. estimer la valeur de la matrice de covariance à la fin de chaque trajectoire
5. calculer  $I[x,a]$  pour chaque trajectoire

MCL représente la probabilité de  $x_t$  par un ensemble de  $M$  particules. La ligne 4 de la méthode ActiveMCL décrite dans l'algorithme 3, tire un échantillon à partir de la probabilité présentée. Le modèle de mesure est appliqué dans la ligne 5 pour déterminer le poids de chaque particule. Le modèle d'implémentation est effectué en utilisant le modèle de perception des antennes RFID. Dans la ligne 3, nous faisons une itération sur toutes les actions possibles de  $t$  à  $t+1$ . Nous obtenons alors les positions suivantes  $x_{t,i}^m$  possibles et les poids  $w_{t,i}^m$ . La probabilité initiale  $bel(x_0)$  est obtenue en générant d'une manière aléatoire  $M$  particules à partir de la distribution à priori  $p(x_0)$  et affectant la fonction d'importance uniforme  $M^{-1}$  à chaque particule. Pour chaque action  $i$ , on calcule la différence dans l'entropie entre la moyenne des particules en  $t$  et  $t+1$ . Dans la ligne 12, nous cherchons l'index de l'action qui maximise  $L(i)$ . Pour évaluer la méthode, nous calculons l'erreur en fonction du nombre des waypoints durant le mouvement du robot. Alors on calcule:

1. La covariance des particules durant le mouvement du robot.
2. La covariance des particules distribuées aléatoirement autour de la vraie position.
3. La différence entre ces deux valeurs.

### 3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES

---

**Algorithm 3** Algorithm ActiveMCL( $X_{t-1}$ ,  $u_t$ ,  $z_t$ ,  $m$ )

---

```

1:  $X_t \leftarrow 0$ 
2: for  $m \leftarrow 1$  to  $M$  do
3:   for  $i \leftarrow 1$  to  $I$  do
4:      $x_{t,i}^m \leftarrow \text{sampleMotionMmodel}(u_{t,i}, x_{t-1}^m)$ 
5:      $w_{t,i}^m \leftarrow \text{measurementModel}(z_t, x_{t,i}^m, m)$ 
6:      $X_{t,i} \leftarrow X_{t,i} + \langle x_{t,i}^m, w_{t,i}^m \rangle$ 
7:   end for
8: end for
9: for  $i \leftarrow 1$  to  $I$  do
10:   $x_{t,i} = \text{mean}(x_{t,i}^m)$ 
11:   $x_{t-1,i} = \text{mean}(x_{t-1,i}^m)$ 
12:   $L(i) \leftarrow \text{entropy}(x_{t,i}) - \text{entropy}(x_{t-1,i})$ 
13: end for
14:  $i_{max} \leftarrow \text{max}(L)$ 
15:  $xa_t \leftarrow x_{t,i_{max}}$ 
16: for  $m \leftarrow 1$  to  $M$  do
17:  draw  $i$  with probability  $w_t^i$ 
18:  add  $xa_t^i$  to  $Xa_t$ 
19: end for
20: return  $Xa_t$ 

```

---

#### 3.4.1 Discussion

Nous montrons dans la figure 3.7 que la prédiction seule ne permet pas une bonne navigation. Ce qui est illustré par l'incertitude qui croît avec le temps.

Nous nous sommes basés sur le paradigme "perception<->action" pour que le robot puisse interagir avec son environnement. Nous remarquons dans cette figure que l'erreur d'incertitude diminue et que les positions s'approchent de la vérité terrain (figure 3.10).

D'autre part, dans la figure 3.11, nous appliquons des amers visuels ponctuels pour la mise

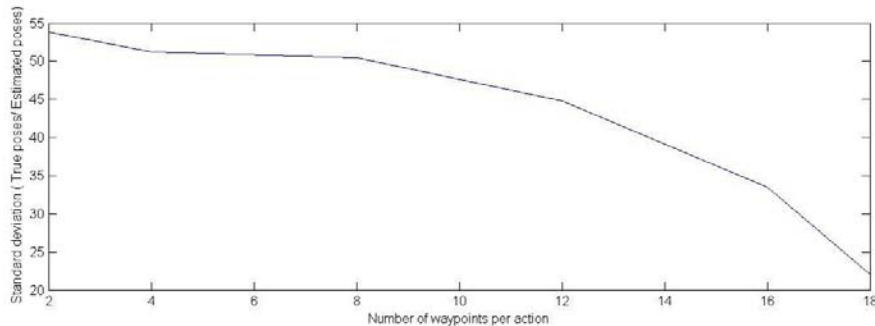


Figure 3.20: Variance de la position en fonction du nombre des waypoints

à jour. Nous utilisons le modèle de sténopé pour avoir les coordonnées de l'amers dans le repère de la caméra. Dans la figure 3.8, nous appliquons l'association de données. Ainsi, le recalage est amélioré par la suppression des particules n'ayant pas les mêmes observations que la vraie position du robot.

La figure 3.8 montre l'amélioration de la précision de la localisation. La variance maximale est de 3.2 cm contre 6.2 cm pour une localisation normale.

La figure 3.20 montre la variation de la variance en fonction du nombre de waypoints.

Notons que la précision de la localisation augmente avec le nombre des waypoints. Par exemple, pour un nombre de waypoints de 4, la variance est 0.52. Pour un nombre de 16, la variance est de 36%.

## 3.5 Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art sur les méthodes de localisation d'un robot fondées sur la technologie RFID. Cette dernière est de plus en plus utilisée dans la navigation. Nous avons ensuite proposé les méthodes de localisation suivantes :

- une localisation déterministe qui utilise l'odométrie et les RFIDs pour recalculer la position d'un robot. Le robot utilise un modèle de perception des tags RFID pour filtrer sa position prédite. Ensuite nous améliorons cette méthode en filtrant les positions qui reçoivent d'autres tags que l'observation.
- une localisation qui utilise des amers ponctuels.
- une localisation qui utilise le modèle Pinhole pour percevoir des points ponctuels. Cela permet au robot de corriger les erreurs dues au bruit.
- une localisation par la méthode Monte Carlo appliquée à la technologie RFID. La position du robot est représentée par un nuage de particules qui sont mises à jour avec un modèle de perception probabiliste.
- enfin, nous proposons une amélioration de cette dernière méthode en utilisant une méthode de localisation active. Elle calcule l'entropie de plusieurs trajectoires possibles que le robot peut emprunter, ensuite elle choisit celle qui maximise cette entropie.

### **3. LOCALISATION DES ROBOTS MOBILES À PARTIR DE RADIO-ÉTIQUETTES**

---

# Localisation et cartographie à partir d'amers visuels

## 4.1 Introduction

Dans ce chapitre, nous traitons le problème de la localisation et de la cartographie simultanée (SLAM) (RGD<sup>+</sup>09). Nous allons développer une méthode de monoslam qui exploite notre représentation par des amers locaux. En effet, la navigation d'un robot mobile dépend de la qualité de la robustesse de sa localisation et de la cartographie de son environnement, d'où notre contribution à étudier ces éléments primordiaux. Compte tenu de la grande variabilité des environnements réels et des limites inhérentes à tout algorithme de localisation et de cartographie, aucune fonction sensori-motrice n'est à même de résoudre à elle seule le problème de la navigation. Nous allons voir comment notre contribution peut effectivement fournir une bonne localisation et une cartographie au système, en complémentarité avec les fonctionnalités déjà implémentées. Nous développons une technique de monoSLAM à base de nos points d'intérêt. Ensuite nous développons une stratégie de navigation en utilisant la technologie RFID et des amers visuels.

## 4.2 Introduction à la localisation et cartographie simultanées

Ce chapitre concerne un des problèmes les plus fondamentaux en robotique, celui du SLAM. Ce problème apparaît lorsqu'on n'a pas accès aux informations sur l'environnement ou à la position du robot. Au lieu de ces informations, nous avons une mesure  $z_i$  et  $n$  déplacements  $u_i$ . Dans le SLAM, le robot acquiert une carte et en même temps il se localise dans la carte.



#### 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

Ce problème est plus difficile que la localisation où la carte est déjà connue. Il est aussi plus difficile que la cartographie où la position du robot est déjà connue.

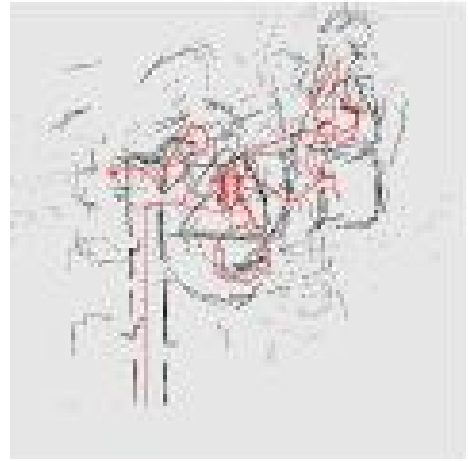
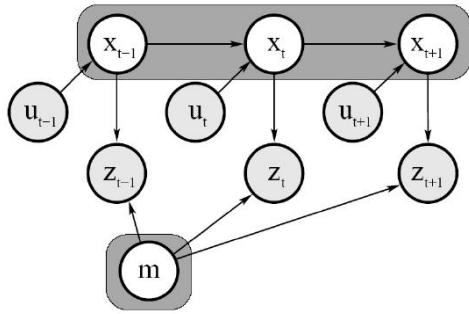


Figure 4.1: Modèle graphique du SLAM en ligne,  $z_i$  étant la mesure du capteur,  $u_t$  l'action,  $x_t$  l'état du robot (à gauche). Une carte construite par un capteur de proximité (à droite)

Dans le problème de SLAM, on s'intéresse simultanément aux coordonnées des amers de la carte ainsi et qu'aux positions du robot. Ces données forment le vecteur d'état du filtre. Dans notre application, nous désirons uniquement des informations relatives à certains points d'intérêt comme les murs et les portes. La position de ces features est estimée dans un référentiel centré sur le robot et la position du robot est explicitement évaluée. L'approche SLAM est attractive car elle encapsule toute l'information importante pour la navigation dans un environnement inconnu. Cependant elle ne peut pas déterminer de quelle distance le robot s'est déplacé de A vers B. Ou même permettre au robot de faire une tâche simple comme retourner à la position de départ après avoir accompli un mouvement particulier. L'approche que nous choisissons estime explicitement la position du robot et les positions des features relatives au repère du monde en calculant les covariances entre toutes les valeurs estimées. La position du robot peut être calculée à partir de quelques features visibles à partir de la position courante. Mais d'autres contraintes comme les occultations ou l'angle maximum de la vue doivent être prises en compte. On utilise généralement le filtre de Kalman (voir l'annexe) où un vecteur contient toutes les quantités ( position du robot et position des features) et une matrice large de covariance qui enregistre les incertitudes relatives.

Dans un problème de SLAM, nous utilisons des observations d'amers et des paramètres de contrôle du robot pour estimer la carte et le chemin du robot. Il y a deux types de représentations (TFB05):

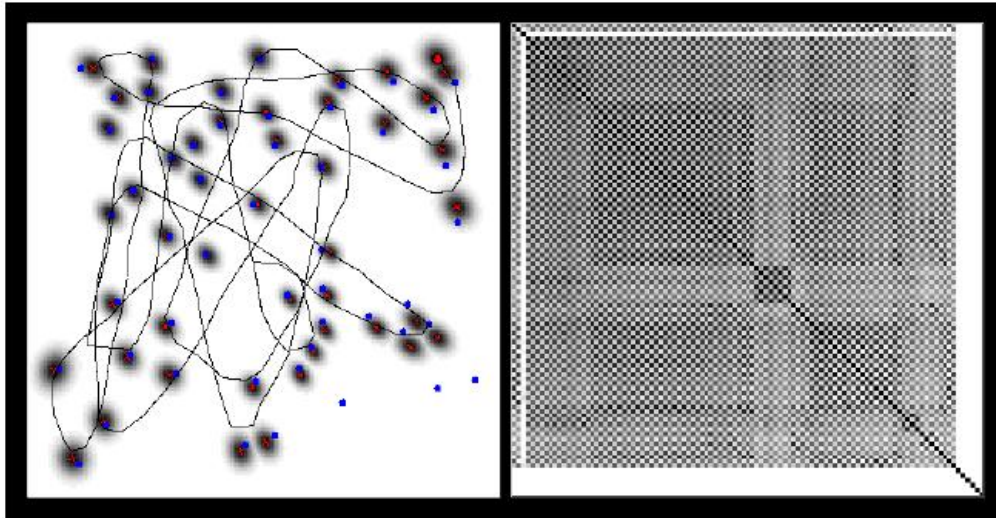


Figure 4.2: Carte (à gauche), Matrice de corrélation (à droite)

- La carte à grille ou le scan
- Les représentations basées sur les amers

Le premier algorithme connu en SLAM date de 1987 ([SSC86](#)), même si la problématique avait été décrite par R.Chatila et JP.Laumond ([CL85](#)) dès 1985. Il a été développé par Smith et Cheeseman. Ils considèrent un monde en 2 dimensions et un robot équipé de capteurs range and bearing (observation par estimation de la distance et de l'angle qui sépare le robot et l'amer). Pour fusionner ces deux types de données, un filtre de Kalman était utilisé. Ainsi, étant donnée une carte de  $N$  amers, on a une gaussienne de  $3+2N$  dimensions qui combine les différents amers entre eux et avec la position du robot. En approche probabiliste, on trouve deux types de problèmes de SLAM :

- Le SLAM en ligne exige l'estimation a posteriori instantanée:

$$P(x_{t;m}/z_{1:t}u_{1:t})$$

où  $x_t$  est la position du robot,

m la carte et  $z_{1:t}$  et  $u_{1:t}$  représentent respectivement les mesures et le contrôle.

- le problème full SLAM : on intègre sur le chemin entier de  $x_1$  jusqu'à  $x_t$  en fonction de la carte au lieu de  $x_t$  uniquement:

$$P(x_{1:t;m}/z_{1:t}u_{1:t})$$

## 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

---



Figure 4.3: Tâches effectuées par un robot pour avoir une représentation correcte de son environnement. Le SLAM est une méthode qui fusionne la localisation et la cartographie

L'EKF SLAM ( SLAM avec l'Extended Kalman Filter) utilise plusieurs approximations

- Les cartes basées sur les amers sont composées d'amers ponctuels. Le nombre est généralement petit. En plus, on obtient des résultats meilleurs lorsque les amers sont moins ambigus ;
- Le bruit gaussien: Comme n'importe quel algorithme de EKF, l' EKF SLAM suppose un bruit gaussien en ce qui concerne le mouvement du robot et la perception.
- La mesure positive : L'algorithme EKF SLAM traite uniquement les observations positives des amers.

Par ailleurs, l'objectif du système SLAM est de garder la densité de probabilité mise à jour lorsque :

- Le robot se déplace ;
- Le robot perçoit un amers qui existe dans la carte ;
- Le robot perçoit un nouveau amer et décide de l'incorporer dans la carte.

### 4.3 Localisation par la vision

#### 4.3.1 Localisation par odométrie visuelle

L'odométrie visuelle utilise une séquence d'images pour estimer le mouvement d'une caméra. (CH00) décrit une odométrie visuelle basée sur l'estimation de la matrice essentielle. À partir

de la correspondance entre plusieurs points dans des images omnidirectionnelles acquises à différentes positions, la matrice essentielle qui minimise les erreurs de reprojection épipolaires est déterminée. L'estimation du mouvement est calculée implicitement, car la matrice essentielle dépend des paramètres du mouvement. Leur méthode a été évaluée dans des environnements extérieurs et intérieurs. (CH00) estime la matrice essentielle à partir d'un ensemble de points extraits d'images omnidirectionnelles acquises à différentes positions. La méthode de (VVS02) est basée sur un modèle de formation d'image applicable sur toutes les caméras à centre de projection unique. On commence par projeter les images sur une rétine sphérique unitaire. Ensuite, on calcule le vecteur vitesse avec une jacobienne relative aux dérivées temporelles sur le système de coordonnées sphériques (Svo99; SPH98) présente une étude de la géométrie sur des images omnidirectionnelles. Il met en correspondance 8 points sur deux images acquises à différentes positions pour estimer la transformation entre les deux acquisitions. (SVS03) effectue une factorisation *multi frame* pour évaluer un mouvement infinitésimal à partir des images omnidirectionnelles. Ils font une dérivation des équations de *flot optique* pour les images panoramiques avec un centre de projection unique. Ces équations sont fonctions des coordonnées du pixel et donc, du rayon de rétro projection associé. On généralise la technique appliquée à la segmentation de mouvement *multi-body* et à l'estimation du mouvement.

#### 4.3.2 Localisation par amers

Les techniques précédentes ne mémorisent aucune référence sur l'environnement. La position estimée est donc entachée d'une erreur qui croît avec la distance parcourue. Dans l'article de ((BZ99)), les auteurs testent statistiquement la fiabilité de la sélection des amers à l'aide d'une série de petits mouvements latéraux du robot en forme d'arcs. Cette stratégie répondant au nom de Turn Back and Look (TBL) est inspirée du comportement chez les guêpes et les abeilles. La figure 4.4 illustre le chemin utilisé par le robot dans la phase TBL de ((BZ99)).

Dans un article de ces mêmes auteurs, une série de petits mouvements latéraux du robot en forme d'arc permet de tester d'une manière statistique la fiabilité de la sélection des amers. Dans les travaux de ((TMZ00)), le robot est éloigné du but tout en gardant la caméra orientée vers lui. Dans une scène qui contient des amers potentiels dans toutes les directions, une scène de TBL est indispensable pour tester la précision des amers. A noter que les mouvements TBL de base sont exécutés sous la forme d'une croix. La précision prise le long de la trajectoire de l'amer est mesurée pour évaluer chaque amer. Le robot s'éloigne de l'origine, ou du but, pendant qu'il garde la caméra. La localisation est effectuée par la mise en correspondance de l'ensemble des amers sélectionnés en chaque endroit par rapport à la scène visuelle courante.

## 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

---

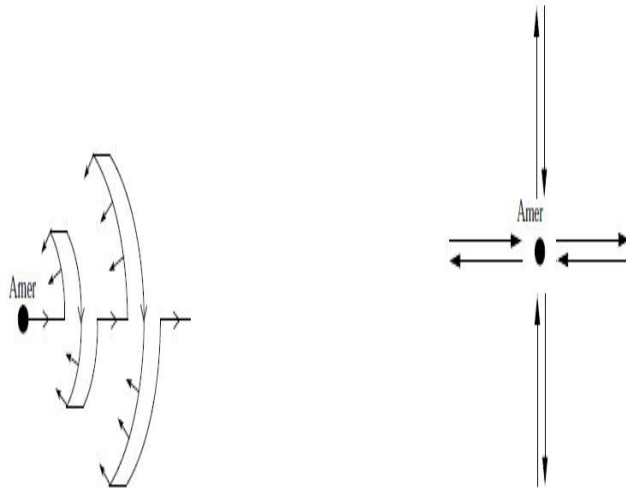


Figure 4.4: TBL, Turn Back and Look (à gauche), TBL dans des images panoramiques (à droite)

D'autres méthodes utilisent des coins extraits comme amers visuels. ((PA02)) utilise quatre points dans l'environnement qui appartiennent à un rectangle plan. Dans un travail de ((DDBC00)), on utilise le capteur SYCLOP pour faire la calibration, la localisation et la reconstruction de cartes en environnement d'intérieur. A partir d'images, ils construisent une carte 2D de l'environnement. Pour cela, ils utilisent des droites verticales de l'environnement comme des portes et des murs. Ces amers ont l'avantage d'être projetés sur les images omnidirectionnelles sous la forme de droites radiales. Après la discrétisation de l'image omnidirectionnelle sous forme angulaire, une mise en correspondance entre les secteurs angulaires est effectuée. Pour la localisation, la méthode proposée fait correspondre les points du modèle sensoriel avec les points répertoriés dans la carte théorique de l'environnement.

### 4.3.3 Localisation par indexation d'images

Afin d'effectuer une localisation, l'information sémantique porte une grande quantité d'information. En effet, les objets qui sont placés autour du robot possèdent des caractéristiques particulières qui leur permettent d'être des amers visuels. Nous notons que les droites sont fréquemment utilisées comme amers dans des environnements intérieurs. Si nous possédons par exemple des caméras panoramiques verticales, alors les droites verticales sont des radiales dans les images panoramiques. Toutefois, nous ne trouvons pas des droites dans les environnements extérieurs naturels. La détection d'amers dans de tels environnements est difficile.

Les méthodes que nous avons présentées dans le chapitre 2 permettent d'extraire des points robustes aux changements d'illumination et de points de vue. Cependant pour faire une bonne

---

## 4.4 Principe et formulation mathématique du MonoSLAM

correspondance, il faut un nombre très important de ces caractéristiques.

Les travaux faits à Oxford par A.Davison et D.Murray ([Dav98](#); [DM98](#); [DM02](#)) ont marqué l'évolution de la navigation par amers visuels. Il ont en effet développé un système monoSLAM dont la fréquence est 30 hz. Neira and al ([NRT97](#)) ont développé quand à eux un système basé sur des segments de ligne verticale appliquée à un environnement intérieur. Dans un autre travail, S.Lacroix et I.K.Jung ([JL03](#)) ont effectué un SLAM en utilisant la stéréo. J.H.Kim et S.Sukkariéh ([KS03](#)) combinent la vision monoculaire avec les capteurs inertiels pour cartographier le sol à partir d'images acquises depuis des robots aériens. Et M.Bosse et P.Newman, encore à Oxford ([BNL+03](#)), ont combiné plusieurs capteurs dans une sous-cartographie qui utilise comme des lignes comme références d'apprentissage. Plusieurs autres méthodes de SLAM ont utilisé le détecteur SIFT. Enfin, Sim and al ([SEGL05](#)) ont combiné le descripteur SIFT et le FastSLAM pour la vision à grande échelle pour la cartographie

## 4.4 Principe et formulation mathématique du MonoSLAM

En matière de SLAM, nous nous intéressons simultanément aux coordonnées des amers de la carte et aux positions du robot. Ces données forment le vecteur d'état du filtre. Dans notre application, nous cherchons uniquement des informations relatives à certains points d'intérêt comme les murs et les portes. La position de ces features est estimée dans des référentiels centrés sur le robot ; ainsi la position du robot est explicitement évaluée. L'approche SLAM est attractive car elle encapsule toute l'information importante pour la navigation dans un environnement inconnu.

L'approche que nous choisissons estime explicitement la position du robot et les positions des caractéristiques relatives au repère du monde en calculant les covariances entre toutes les valeurs estimées. La position du robot peut être calculée en utilisant quelques features visibles à partir de la position courante. Mais d'autres contraintes comme les occultations ou l'angle maximum de la vue doivent être prises en compte.

Les positions du robot et des features de la scène sont stockées dans le vecteur d'état du système et l'incertitude des estimées est stockée dans la matrice de covariance  $P$ .

### Construction de la carte

Nous utilisons une carte probabiliste basée sur nos points d'intérêt qui représente dans chaque intervalle de temps, l'estimée de l'état de la caméra et tous les points d'intérêt extraits. Notre système est initialisé et est mis à jour en fonction du mouvement de la caméra et de l'observation des points d'intérêt.

#### 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

Le caractère probabiliste de la carte est dû à la propagation de l'estimée de l'état de la caméra et des points d'intérêt et aussi à la distribution des incertitudes premier ordre.

Soient les positions du robot et des features de la scene sont stockées dans le vecteur d'état du système et l'incertitude des estimées est stockée dans la matrice de covariance P.

Elles sont exprimées comme suit :

$$\hat{x} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \end{pmatrix}, P = \begin{pmatrix} P_{xx} & P_{xy_1} & P_{xy_2 \dots} \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2 \dots} \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2 \dots} \end{pmatrix} \quad (4.1)$$

Où  $\hat{x}$  a  $3(n+1)$  éléments, n étant le nombre des features connus. P est une matrice symétrique avec une taille de  $3(n+1) \times 3(n+1)$  ;  $\hat{x}$  et P vont changer en dimension vu que les features sont additionnés ou supprimés de la carte.  $\hat{x}_v$  est la position estimée du robot et  $\hat{y}_i$  la position 3D estimée du ième feature.

$$\hat{x}_v = \begin{pmatrix} \hat{z} \\ \hat{x} \\ p\hat{h}_i \end{pmatrix}, \hat{y}_i = \begin{pmatrix} \hat{X}_i \\ \hat{Y}_i \\ \hat{Z}_i \end{pmatrix} \quad (4.2)$$

Remarque: ces coordonnées sont toutes exprimées dans le repère du monde W.

##### Initialisation du filtre :

Nous initialisons le système avec  $x(1) = 0$ ,  $x(2) = 0$ ,  $x(3) = 0$ . La matrice de covariance de départ a toutes les entrées égales à zéro.

$$\hat{x}_v = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, P = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

##### Mouvement et prédiction du mouvement

Le mouvement du robot est discrétisé en étapes d'intervalles de temps, avec un facteur d'incrément k. Après une étape de mouvement, une nouvelle estimation de l'état et de covariance est produite.

$$\hat{x}(k+1|k) = \begin{pmatrix} f_v(X_v(k|k), u(k)) \\ \hat{y}_1(k|k) \\ \hat{y}_2(k|k) \end{pmatrix}$$

$$P(k+1|k) = \begin{pmatrix} \frac{df_v}{dx_v} P_{xx}(k|k) \frac{df_v^T}{dx_v} + Q(k) & \frac{df_v}{dx_v} P_{xx}(k|k) & \frac{df_v}{dx_v} P_{xy_2}(k|k) \dots \\ P_{y_1x}(k|k) \frac{df_v^T}{dx_v} & P_{y_1y_1}(k|k) & P_{y_1y_2}(k|k) \dots \\ P_{y_2x}(k|k) \frac{df_v^T}{dx_v} & P_{y_2y_1}(k|k) & P_{y_2y_2}(k|k) \dots \end{pmatrix}$$

#### 4.4 Principe et formulation mathématique du MonoSLAM

Où  $f_v$  et  $Q(k)$  sont respectivement le modèle de déplacement et la covariance sur le déplacement

Cette nouvelle matrice de covariance est formulée à partir de la règle de prédiction EKF suivante :

$$P(k+1|k) = \frac{df_v}{dx_v} P(k|k) \frac{df_v^T}{dx_v} + Q(k)$$

Nous avons aussi,

$$\frac{df}{dx} = \begin{pmatrix} \frac{df_v}{dx_v} & 0 & 0 & \dots \\ 0 & I & 0 & \dots \\ 0 & 0 & I & \dots \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \end{pmatrix} \quad (4.3)$$

##### Mesure d'un amer visuel

Faire une mesure d'un feature  $i$  consiste à déterminer sa position  $h_g$  relative au robot .

$$h_{G_i}^{C0} = \begin{pmatrix} h_{G_i}^{C0x} \\ h_{G_i}^{C0y} \\ h_{G_i}^{C0z} \end{pmatrix} = \begin{pmatrix} \cos(\phi(X_i - x)) - \sin(\phi(Z_i - z)) \\ Y_i - H \\ \sin(\phi(X_i - x)) + \cos(\phi(Z_i - z)) \end{pmatrix} \quad (4.4)$$

La mesure prédite d'un feature à la position courante du robot est calculée à partir des estimées d'états  $\hat{x}_v$  et  $\hat{y}_i$ , substitué par  $x_v$  et  $y_i$ . Quand une mesure de ce vecteur est faite, la covariance de l'innovation ( qui est la covariance de la différence entre les valeurs vraies et mesurées) est donnée par :

$$S_{h_{G_i}} = \frac{dh_{G_i}}{dx} P \frac{dh_{G_i}^T}{dx} + R_L = \frac{dh_{G_i}}{dx_v} P_{xx} \frac{dh_{G_i}^T}{dx_v} + 2 \frac{dh_{G_i}}{dx_v} P_{xy_i} \frac{dh_{G_i}^T}{dy_i} + \frac{dh_{G_i}}{dy_i} P_{y_i y_i} \frac{dh_{G_i}^T}{dy_i} + R_i. \quad (4.5)$$

Pour mesurer le point d'intérêt ou feature, l'angle de la caméra est dirigé vers sa position. Considérons la projection de l'image sur la caméra :

$$u = -f k_u \frac{h_x}{h_z} + u_0$$

$$v = -f k_v \frac{h_y}{h_z} + u_0$$

La matrice de covariance du vecteur image  $u_L = \begin{pmatrix} u_L \\ v_L \end{pmatrix}$  est donnée par :



#### 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

---

$$U = \frac{du}{dh} P_h \frac{du^T}{dh}$$

La valeur du jacobien à  $h_{Lx}^L = h_{Ly}^L = 0$  est :

$$\frac{du_L}{dh_L^L} = \begin{pmatrix} -\frac{f.k_u}{h_{Lz}} & 0 & 0 \\ 0 & -\frac{f.k_v}{h_{Lz}} & 0 \end{pmatrix} \quad (4.6)$$

##### Mise à jour du vecteur d'état après une mesure

Chaque mesure est transformée en sa forme angulaire avant son traitement :

$$h_i = \begin{pmatrix} \alpha_i \\ e_i \\ \gamma_i \end{pmatrix} = t \begin{pmatrix} \tan^{-1} \frac{h_{Gix}}{h_{Giz}} \\ \tan^{-1} \frac{h_{Giy}}{h_{Giz}} \\ \tan^{-1} \frac{I}{2h_{Gi}} \end{pmatrix} \quad (4.7)$$

Pour réaliser une bonne estimation de la position de la caméra, nous avons besoin d'amers visuels stables, valables à long terme, identifiables et saillants. Nous utilisons ainsi les amers que nous avons développés dans le chapitre 2. Ils sont en effet invariants à la rotation et aux changements d'échelle. Ils permettent aussi de décrire les images en couleur. Nous allons les exploiter pour construire la carte du robot et pour le localiser.

Nous utilisons l'approche de A.Davison (Dav98) pour définir le modèle de caméra de mouvements. Nous définissons le système de coordonnées  $W$  fixé dans le monde et  $R$  fixé à la caméra. On choisit une représentation non minimale de l'orientation 3D et on utilise un quaternion. Le vecteur de 7 paramètres est utilisé pour représenter une position et une orientation.

$$x_p = \begin{pmatrix} r^w \\ q^{WR} \end{pmatrix} = (xyzq_0q_xq_yq_zq_z)^T$$

Où  $q$  définit l'orientation,

Dans le cas des systèmes déterministes, un système fermé peut être modélisé avec une précision infinitésimale par un ensemble de paramètres et équations et alors le comportement futur est totalement prédit.

Toutefois en réalité, la précision du modèle s'arrête à un certain degré. Dans le cas d'un robot à roues, ce terme de bruit prend compte de facteurs comme le patinage des roues, les irrégularités des surfaces. Cependant, notre modèle considère que la vitesse du robot est constante. Ce qui ne veut pas dire que la vitesse est constante mais que l'accélération a un profile gaussien.

## 4.5 MonoSLAM par nos points d'intérêt

---

Ceci implique que le moment de la caméra possède une forme lisse. Cette propriété donne ainsi au système une certaine robustesse même si les mesures visuelles sont éparses. Pour modéliser la vitesse de la caméra avec cette méthode, nous devons ajouter le terme de la vitesse à la position dans le vecteur d'état.  $V^w$  est la vitesse linéaire et  $W^w$  est la vitesse angulaire. La dimension totale du vecteur d'état de la caméra est 13.

$$x_v = \begin{pmatrix} r^w \\ q^{WR} \\ v^W \\ \omega^W \end{pmatrix}$$

On assure qu'à chaque étape une accélération inconnu  $a^w$  et une accélération angulaire  $\alpha^w$  ont une moyenne nulle et que la distribution gaussienne cause une impulsion de vitesse et une vitesse angulaire.

La mise à jour produite de l'état est :

$$f_v = \begin{pmatrix} r_{new}^w \\ q_{new}^{WR} \\ v_{new}^W \\ \omega_{new}^W \end{pmatrix} = \begin{pmatrix} r^w + (v^w + V^w)\delta_t \\ q^{WR} * q((\omega^W + \Omega^W))\delta_t \\ v^W + V^W \\ \omega^W + \Omega^W \end{pmatrix}$$

La notation  $v^w + V^w\delta_t$  indique le quaternion défini par l'axe d'angle du vecteur de rotation. Dans l'EKF, l'estimée du nouvel état  $f_v(x_v, u)$  est accompagné par l'augmentation dans le vecteur d'incertitude  $Q_v$  pour la caméra après le mouvement. Nous obtenons  $Q_v$  via le calcul du jacobien où  $P_n$  est la covariance du vecteur de bruit n.

## 4.5 MonoSLAM par nos points d'intérêt

La méthode consiste à parcourir une surface et à en extraire l'ensemble des images par une monocaméra. Ensuite, pour chaque image, on extrait les points d'intérêt pour effectuer un appariement avec l'image précédente. On garde par conséquent les features appariés. On calcul aussi la covariance de ces points d'intérêt. On prédit donc la position suivante du robot en utilisant le modèle de l'odométrie et on calcule la mesure des points d'intérêt. Cela nous permet de calculer la nouvelle position mise à jour du robot et les positions des points d'intérêt via le filtrage EKF.

L'algorithme 4 présente notre méthode. Nous l'avons implémenté sous MATLAB.

## 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

---

**Algorithm 4**

---

```
1: for  $i \leftarrow 1$  to  $Nombre - image$  do  
2:   lire image  
3:    $pos(i) \leftarrow$  Extract-features  
4:    $match(pos(i), pos(i-1))$   
5:   Ajouter  $pos(i)$  à  $x_k$  ( vecteur d'état)  
6:    $calculer-covariance(p_k)$   
7:   prédire  $x_k(i + 1)$   
8:   mesurer les features  $pos_i$   
9:    $mise-a-jour(x_k, p_k)$   
10: end for
```

---

### 4.5.1 Simulation

#### Génération des caractéristiques des images de l'environnement

Les positions clés sont sélectionnées en utilisant la méthode développée en vision. Cette technique permet d'avoir des points d'intérêts invariants à l'échelle et à la rotation et assure la description de la couleur. Les descripteurs sont particulièrement stables pour caractériser l'image. Après l'appariement des descripteurs entre les frames, nous obtenons une estimation du mouvement de la caméra à partir de l'odométrie du robot pour prédire les caractéristiques des descripteurs dans la nouvelle frame. Afin de construire une carte, nous avons besoin de savoir comment le robot a bougé entre les frames pour mettre les amers ensemble d'une manière cohérente. Nous fusionnons pour cela les données de l'odométrie et la carte construite pour déterminer une localisation plus précise de la caméra. Une fois la position précise, on peut déterminer la position des amers dans la nouvelle image. Le matching entre deux frames est effectué en calculant le produit entre les vecteurs des descripteurs dans deux frames successives. Nous gardons seulement les appariements dans lesquels le ratio de l'angle entre les deux vecteurs les plus proche est inférieur que 0.1.

## 4.6 Navigation par fusion entre radio-étiquettes et points d'intérêt

Après avoir appliqué notre méthode d'extraction de points d'intérêt pour la localisation et la cartographie simultanée, nous avons développé une stratégie de navigation en fusionnant nos amers visuels et des tags RFID. Le principe de notre algorithme est de recalculer en utilisant un filtrage particulière (rappelé en 5.6), comme décrit dans la méthode probabiliste présentée en chapitre 3. Si le robot ne détecte aucun tag, alors il fait sa correction en utilisant les points d'intérêt extraits. L'algorithme 5 donne sous la forme d'un pseudo-code, les principales étapes de notre méthode.

#### 4.6 Navigation par fusion entre radio-étiquettes et points d'intérêt



Figure 4.5: Résultats d'estimation de la position des patches des descripteurs pour une scène intérieure

#### 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

---

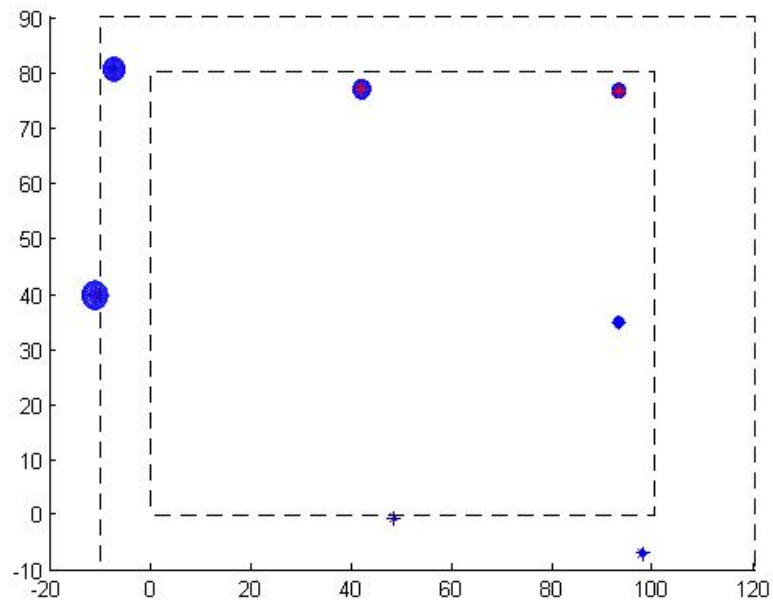


Figure 4.6: Position prédite de la caméra avec les incertitudes en utilisant l'odométrie uniquement

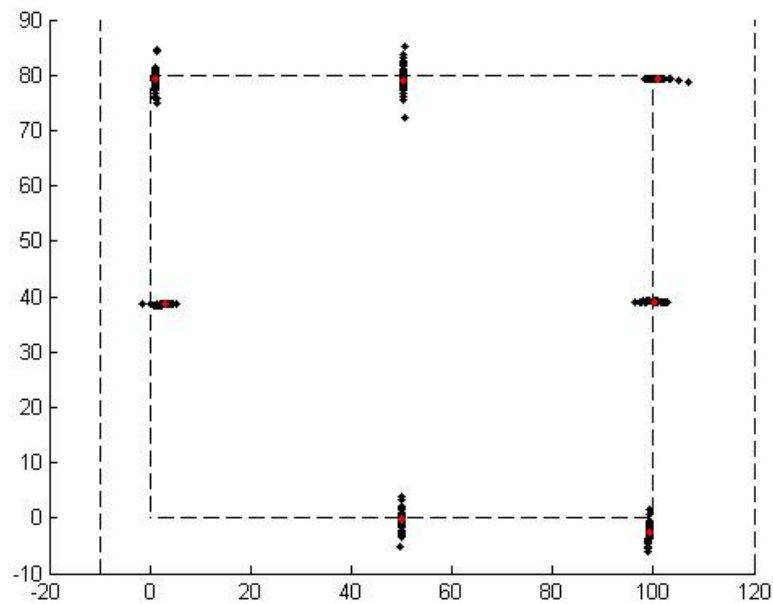


Figure 4.7: Position estimée du robot en utilisant l'odométrie et les amers visuels

**Algorithm 5**


---

```

1: for  $i \leftarrow 1$  to  $N$  do
2:   if reception RFID then
3:     Prediction()
4:     Correction()
5:     Re-echantillonnage()
6:   else
7:     lire image
8:      $\text{pos}(i) \leftarrow \text{Extract-features}$ 
9:      $\text{match}(\text{pos}(i), \text{pos}(i-1))$ 
10:    add  $\text{pos}(i)$  to  $x_k$  ( vecteur d'état)
11:     $\text{calculer-covariance}(p_k)$ 
12:    predire  $x_k(i + 1)$ 
13:    mesurer les features  $\text{pos}_i$ 
14:    mettre à jour( $x_k, p_k$ )
15:   end if
16: end for

```

---

La figure 4.6 montre les positions prédites en utilisant le modèle de l'odométrie  $P(x_{t+1}/x_t, u)$ . Nous remarquons que l'incertitude sur la position augmente avec le temps. Ceci est dû à l'accumulation de l'erreur sur la position du robot; la navigation du robot échoue parce qu'elle se base uniquement sur le modèle de l'odométrie. La covariance de la position croît avec le temps. Plusieurs facteurs peuvent causer les erreurs comme le glissement ou le rebondissement des roues.

Dans la figure 4.7, nous appliquons notre méthode de correction de la position en nous basant uniquement sur les RFIDs; nous montrons une simulation qui utilise uniquement la méthode Monte-Carlo avec les RFID (chapitre 3). Ceci permet de réduire l'incertitude sur la position estimée. Par conséquent, nous nous approchons de la vérité terrain.

Enfin, dans la figure 4.6, nous combinons deux capteurs, la caméra et les antennes RFID, en exploitant les amers visuels pré-calculés (voir sections précédentes de ce chapitre). Nous remarquons que l'erreur sur la position diminue par rapport aux figures 4.6 et 4.7. Notre stratégie consiste à percevoir les tags RFID. Si aucune observation n'est faite, le robot utilise sa caméra.

## 4.7 Conclusion

Dans ce chapitre, nous avons appliqué nos points d'intérêt au cas du MonoSLAM. Cette dernière est utilisée fréquemment dans la navigation car elle combine en un seul algorithme la localisation et la cartographie. Nous démontrons à travers une simulation que notre algorithme

#### 4. LOCALISATION ET CARTOGRAPHIE À PARTIR D'AMERS VISUELS

---

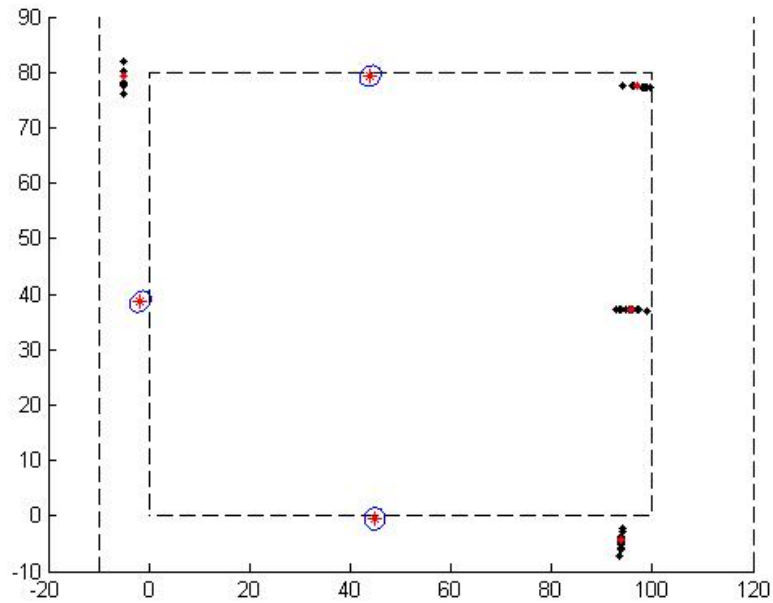


Figure 4.8: Recalage des positions du robot par fusion des RFIDs et des amers visuels

réduit l'incertitude sur la position du robot. Notre deuxième méthode consiste à combiner en même temps les données des capteurs visuels et RFID. En effet, nous avons remarqué suite à nos expériences sur les RFID que dans plusieurs cas, le robot ne détecte pas de tags. Par conséquent, nous développons une stratégie de navigation qui localise le robot par nos amers visuels s'il ne reçoit pas des tags RFID.

# Cartographie des radio-étiquettes

## 5.1 Introduction

Le robot doit être capable de modéliser son environnement grâce à l'ensemble de ses capteurs pour pouvoir se déplacer et agir. Ce modèle est nécessaire pour des tâches de localisation ainsi que pour la planification de mouvements et l'accomplissement de missions. Par exemple, les données qui proviennent d'images très précises (satellites par exemple) ont malgré tout une précision insuffisante pour des applications qui ont besoin de perceptions précises dans une zone d'activité locale du robot. En plus, le robot doit être flexible pour prendre en compte des modifications inattendues de son environnement.

Par conséquent, la construction de la carte doit être incrémentale en fusionnant les perceptions successivement acquises par le robot durant son déplacement.

Lors de la construction de la carte du robot, plusieurs difficultés apparaissent :

- la taille de la carte construite : En effet, plus l'environnement décrit par une carte est grand, plus l'espace mémoire pour la stocker est grand et plus les calculs sont lourds ;
- le bruit qui entache les mesures des capteurs et les mouvements effectués par les actionneurs rend la cartographie difficile ;
- l'ambiguïté perceptuelle : Cela est dû au fait que différents endroits de l'environnement peuvent avoir des apparences très similaires: cirons par exemple des couloirs dans un grand bâtiment (hotel, hopital...). Il est difficile de faire des correspondances entre des points extraits dans ces endroits traversés avec des instants différents. En d'autres termes, le problème de l'association de données doit être traité ;



## 5. CARTOGRAPHIE DES RADIO-ÉTIQUETTES

---

- fermeture de la boucle: quelquesoit la méthode utilisée pour intégrer les données sur la position du robot, les erreurs augmentent avec la distance parcourue. Le SLAM permet de corriger les erreurs odométriques qui s'accumulent par intégration. Si aucun amer n'est revu pendant le déplacement pour localiser le robot, la carte construite incrémentalement peut se déformer. Pour obtenir une carte cohérente, on doit reconnaître des amers vus en une position par le passé et utiliser une technique de SLAM pour construire la carte, soit par des approches d'estimation (filtrage probabiliste), soit par des approches d'optimisation.

### 5.2 Modèles géométriques

Le modèle contient des informations métriques sur l'environnement. Dans cette représentation, nous utilisons l'ensemble des primitives 2D acquises par un balayage horizontal d'un télémètre laser. Ainsi, par la vision, on utilise des primitives 3D comme les segments regroupés en facettes, décrivant ainsi les amers dédiés à la localisation et représentant l'espace navigable. La sélection des amers doit répondre aux critères suivants : pouvoir discriminant, domaine de visibilité, invariance et bonne adaptation à la météorologie. Il existe plusieurs types d'amers géométriques comme les points, les coins, les segments et les polygones.

- Les points ou objets ponctuels : Dans les cartes 2D, ils représentent notamment des objets verticaux tels que des troncs d'arbre ou des poteaux ou des coins de mur.
- Le coins : comparables aux points, cependant, ils contiennent une information d'orientation et un angle relatif entre les segments adjacents à ce coin.
- Les segments : qui sont associés aux frontières d'obstacles. Ils sont représentés par leurs droites support (en coordonnées polaires de préférence) et portent éventuellement des informations géométriques de leurs extrémités.
- Des frontières d'obstacles dans le plan horizontal représenté par des polygones de forme et de taille quelconque.
- Position de courbe, en particulier avec des cercles.

Les représentations géométriques sont exploitées dans la description des environnements naturels avec des terrains plutôt plats. Elles sont aussi utilisées pour modéliser des environnements structurés. Par exemple, des cartes géométriques incluant des coins et des surfaces partielles, ou aussi des cartes construites par des approches récursives pour segmenter l'environnement en utilisant le sonar.

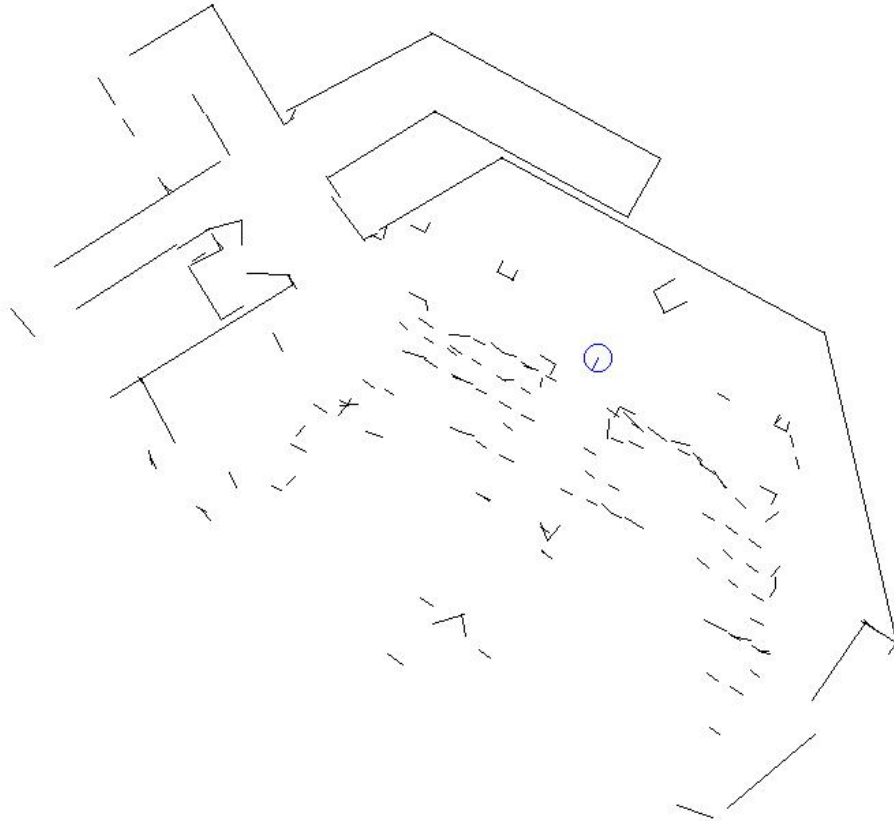


Figure 5.1: Exemple d'une carte géométrique acquise dans une salle de la Cité de l'Espace à Toulouse, à partir d'un télémètre laser plan.

Différents types d'objets peuvent être inclus dans une carte géométrique. Les données métriques brutes sont produites via une fusion successive des primitives comme les points. Ceci permet la réduction de l'espace mémoire nécessaire pour stocker la carte. Toutefois, il s'avère qu'une carte géométrique est plus difficile à construire comparativement à une grille d'occupation. Ceci est dû en particulier aux approximations nécessaires. De plus, pour effectuer une reconnaissance d'objet ou identifier des formes géométriques utiles pour la localisation du robot, il faut mettre en oeuvre des traitements de mise en forme pour l'ensemble des données quand le robot effectue un raisonnement sur sa propre carte.

### 5.2.1 Cartes métriques

Dans un espace métrique l'environnement est associé à un ensemble d'objet qui correspondent à des positions dans un espace métrique. Nous utilisons la position du robot estimée pour déterminer celle des objets dans l'environnement via les perceptions. Dans les cartes métriques, nous utilisons la fusion de plusieurs sources d'informations. L'avantage de ces

## 5. CARTOGRAPHIE DES RADIO-ÉTIQUETTES

---

cartes est la possibilité d'avoir une carte exhaustive des positions que le robot peut explorer; elle s'étend à tous les lieux que le robot a pu percevoir à partir des zones qu'il a visitées. Aussi, les cartes métriques permettent d'avoir une position exacte du robot. Il s'ensuit une utilisation directe et simple. Une telle représentation permet une réutilisation de la carte pour d'autres capteurs à condition qu'on ait le modèle du capteur. Cette représentation permet de faire appel à des représentations de plus haut niveau tel que des objets, des obstacles ou des murs. L'inconvénient des cartes métriques est qu'elles sont fortement liées aux données proprioceptives ce qui nécessite une odométrie plus fiable. On peut réaliser cela en faisant des restrictions sur l'environnement. Par exemple on peut considérer que tous les couloirs sont orthogonaux pour éviter des erreurs sur la position du robot.

### **Implantation**

L'implantation est effectuée en utilisant deux approches. Soit qu'on utilise directement des objets extraits par différentes manières et de différents types d'objets soit on utilise directement l'espace libre sans avoir recours à identifier l'objet.

*Objets points:* Les objets peuvent être des points. Cette méthode n'est pas très fiable car un point ne permet pas d'avoir une bonne localisation de l'objet. Aussi on peut utiliser un ensemble de points disséminés sur l'objet, dans ce cas on utilise des télémètres laser pour faire la perception. *Objets points orientés.* on peut doter un objet point d'une orientation, ceci permet d'avoir une position unique du robot. Dans ce cas, un objet complexe est identifiable par ce point orienté, par exemple l'angle d'un obstacle perçu par un télémètre laser.

**Frontières des objets :** Les frontières des objets et les obstacles peuvent aussi être représentés par différentes formes géométriques, des lignes et des polygones peuvent aussi être utilisés. Les points sur ces objets sont identifiés par des télémètres laser ou des capteurs à ultrason ou aussi par des caméras stéréo pour les objets 3D.

### 5.3 Modèle topologique

L'espace navigable du robot est représenté sous forme d'un graphe contenant toutes les informations sur la connexité et l'ordre des entités incluses. Autrement, il s'agit d'une représentation décrivant les relations entre des éléments potentiels de l'environnement sans utiliser une référence globale absolue ni une métrique en générale. Pour construire un modèle topologique, on doit :

1. construire un modèle géométrique, ensuite chercher ses propriétés topologiques. Dans ce modèle, la définition du noeud est liée à une unité fonctionnelle ou topologique (couloir, bureau) et les connecteurs entre les noeuds représentants des portes ou des escaliers;

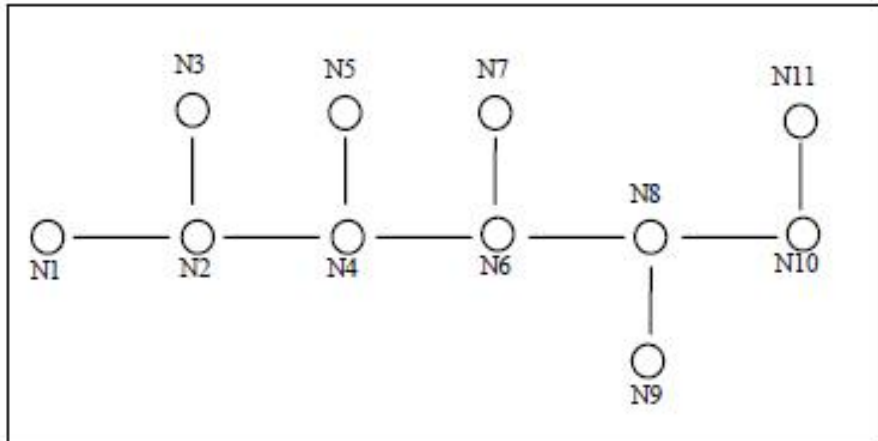


Figure 5.2: Carte topologique

2. utiliser des informations qualitatives ou physiques comme des couleurs ou des textures afin de distinguer la nature physique d'une partie de l'environnement traversé pour la considérer comme une entité discrète dans le graphe et aussi pour rajouter éventuellement des informations métriques locales;
3. considérer les représentations topologiques inhérentes aux environnements structurés (pièces, portes...), car elles induisent une structure topologique avec moins d'ambiguïté. L'un des avantages des représentations topologiques est d'éviter les approximations géométriques par des primitives tout en rendant compte de la structure de l'environnement;
4. exploiter la structure de la carte topologique dans des algorithmes de planification de mouvements.

Un modèle topologique est une représentation économe en espace mémoire. Cependant, l'absence d'informations géométriques peut être inadéquate dans les cas où la localisation précise est indispensable. De même, si plusieurs parties de l'environnement ont des apparences presque semblables, une ambiguïté de positionnement du robot peut apparaître.

### 5.3.1 Cartes topologiques

La perception intervient dans la détection et la mémorisation des lieux. Alors que la première permet de comparer deux perceptions et donc de reconnaître un lieu de la carte ou de détecter un lieu nouveau, la seconde procédure permet de mémoriser un nouveau lieu ou d'adapter la définition d'un lieu dans des passages successifs du robot en ce lieu. Mais, du moment que la reconnaissance d'un lieu est soumise au problème de la variabilité perceptuelle, la première

## 5. CARTOGRAPHIE DES RADIO-ÉTIQUETTES

---

procédure peut donner des résultats erronés. Il est à noter que les données mémorisées sur les arêtes, des relations de voisinage proviennent des données proprioceptives. Ces types de données sont en général caractéristiques des cartes topologiques où la position du robot n'est pas importante mais c'est plutôt reconnaître un lieu qui importe. Ces données peuvent être des informations sur la position relative du noeud ou sur les actions à effectuer pour parcourir cette arête. L'avantage des cartes topologiques est qu'elles n'ont pas besoin de données métriques pour représenter l'environnement en fusionnant les données proprioceptives et les perceptions. Concernant l'erreur produite par cette procédure, elle est minimisée car il devient plus facile de considérer un bruit émanant en même temps des éléments proprioceptifs et des capteurs.

La mémorisation sous forme d'un ensemble de lieux distincts autorise en général une définition des lieux plus directement reliée aux capacités perceptives du robot.

La propriété de cette représentation avec les données brutes conduit en général à utiliser moins de concepts de haut niveau que la représentation métrique. Donc la carte topologique reste proche du robot en mémorisant ses concepts et ses déplacements indépendamment des concepts de plus haut niveau.

Un autre point fort des cartes topologiques est la discrétisation de l'environnement relativement aux choix des lieux dans la carte. Cette discrétisation permet en effet de réduire la planification en une simple opération de recherche dans un graphe.

### **Inconvénient:**

Utiliser les perceptions sans modèle métrique empêche d'estimer la position du robot dans un repère de référence? que cela soit sur une zone visitée ou non visitée. Donc, les cartes topologiques nécessitent un mode de commande du robot, qui n'exploite pas la localisation, typiquement le suivi de trajectoire planifiée à partir d'une carte métrique, n'est pas possible. De ce fait la navigation doit être fondée sur la commande référencée capteur, par exemple sur l'asservissement visuel (par exemple avec des fonctions comme *Suivre le mur*, *Franchir la porte*, *Suivre le couloir*...).

### **Implantations:**

*Définition des noeuds :*

Le choix des noeuds est d'une grande importance dans les cartes topologiques. Il est lié aux capacités perceptuelles du robot qui devra détecter les lieux en questions.

- Noeuds définis par le concepteur :

Le concepteur peut déterminer manuellement quels endroits doivent être détectés par le robot et comment ces endroits doivent être représentés et reconnus. Le choix du type de lieu est détecté par des procédures. Le choix le plus courant est l'utilisation de couloirs de portes et d'intersections. Avec ce choix, un nombre petit d'endroits peut être détecté ce qui laisse le problème du perceptuel aliasing omniprésent. Cela donne une importance aux informations proprioceptives.

- Noeuds définis à des positions canoniques :

Le concepteur précise dans quels types de situations le robot doit enregistrer les lieux, laissant le soin de définir chaque lieu au moment de la découverte. Par exemple, il peut enregistrer des noeuds lorsqu'il détecte des portes par la couleur de la porte ou le numéro inscrit sur la porte.

- Noeuds définis de manière non supervisés:

Dans cette méthode, on ne considère que les noeuds dont des zones ou les perceptions sont approximativement constantes. Cela est obtenu par la catégorisation non supervisée de perception. Les perceptions sont regroupées en données similaires sans que ces catégories soient spécifiées par un concepteur humain. Cette méthode s'avère bien adaptée à des robots autonomes car la catégorisation ne nécessite aucun superviseur ni aucune définition a priori des données correspondant à un noeud. Elle est ainsi utilisée pour tous les systèmes de navigations s'inspirant des comportements de navigation des animaux.

A ce niveau, il est nécessaire de définir un critère pour décider quand un nouveau lieu a été atteint. Il est plus évident de comparer constamment la situation courante à celle du noeud précédent reconnu. Si la différence est suffisamment grande, on considère qu'un nouveau lieu est atteint.

#### *Définition des arêtes:*

les arêtes sont utilisées pour la mémorisation des relations de voisinage entre les noeuds. Les données sont des informations proprioceptives. Elles peuvent être plus ou moins précises et se présentent sous diverses formes.

#### **Relation d'adjacence:**

Elles portent les informations d'adjacence entre les deux noeuds qu'elles connectent. Elle peuvent être bidirectionnelles ou non. Elles permettent au robot de passer d'un lieu à l'autre.

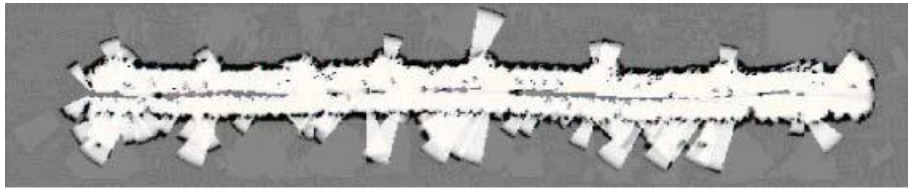


Figure 5.3: Grille d'occupation

### 5.4 Grille d'occupation

On décompose l'espace où le robot mobile évolue en un ensemble de cellules. A chaque cellule est associée une probabilité d'occupation avec une valeur estimée à partir des mesures fournies par les capteurs.

Dans les approches où l'on fait du SLAM, on considère que les environnements d'un robot mobile sont statiques. Les grilles d'occupation permettent de suivre l'évolution de l'environnement en mettant à jour à une fréquence élevée les probabilités d'occupation. En outre, les grilles d'occupation permettent de modéliser des environnements de formes diverses ce qui évite de chercher à approximer les données par des primitives qui peuvent être inadéquates. On peut utiliser cette approche, par exemple, sur des applications qui détectent les espaces navigables (espaces libres) ou pour l'évitement d'obstacles. Par contre, la discrétisation par une grille induit des déformations et nécessite un espace de stockage important.

Cette méthode a été introduite par A.Elfe à CMU ([Elf92](#)); il a proposé une représentation discrète 2D de l'espace navigable par un robot mobile à l'aide des capteurs à ultrasons. L'utilisation de cette représentation décompose l'environnement modélisé en cellules. A chaque cellule, on assigne un vecteur d'état  $X_t$  qui indique l'état d'occupation de cette cellule à l'instant  $t$ .

La méthode repose sur l'utilisation du filtre bayésien discret (DBF) applicable normalement aux problèmes avec des espaces d'états finis. On exploite les mesures reçues du capteur pour produire une prédiction du changement de l'état de la cellule. Ci après, les deux étapes de ce filtre :

Soient les trois croyances  $X_t^{lib}$ ,  $X_t^{occ}$ ,  $X_t^{inc}$  et soit  $z_t$  une mesure reçue par le capteur à un instant  $t$

**Etape de prédiction :**

$$\hat{P}(X_t^{occ}) = P(X_t^{occ}|z_t, X_{t-1}^{occ}) * P(X_{t-1}^{occ}) + P(X_t^{lib}|z_t, X_{t-1}^{lib}) * P(X_{t-1}^{lib}) + P(X_t^{inc}|z_t, X_{t-1}^{inc}) * P(X_{t-1}^{inc})$$

**Etape de correction :**

## 5.5 Cartographie d'un environnement par des position de radio-étiquettes

---

### Algorithm 6

---

```
1: for tag ← 1 to N do
2:   for robot-position ← 1 to P do
3:     detected-tags = scan(environment)
4:     if detected-tags ≠ ∅ then
5:       memorize this zone  $z_{robot-position}$ 
6:       intersect with the precedent  $z_{robot-position-1}$ 
7:     end if
8:   end for
9: end for
```

---

$$P(X_t^{occ}) = P(X_t^{occ}|z_t) = P(z_t|X_t^{occ}) * \hat{P}(X_t^{occ}) * P_{dist}$$

## 5.5 Cartographie d'un environnement par des position de radio-étiquettes

Dans notre application, nous cherchons les positions du robot en appliquant la localisation par vision ou tout autre méthode: en effet, il paraît délicat de traiter complètement du SLAM uniquement par des RFIDs. Ensuite, nous construisons notre carte de radio-étiquettes. Nous proposons deux méthodes: l'une est déterministe et l'autre est probabiliste.

### 5.5.1 Méthode déterministe

On considère que le robot est muni d'antennes quand il se déplace dans l'environnement, où les tags sont placés. La méthode est décrite par le pseudo-code présenté dans l'algorithme 6. A chaque fois qu'il détecte un tag, il réduit la zone où le tag peut exister. Supposons qu'il soit dans la zone A 5.4, il trace sa zone d'existence. Après le déplacement, la zone éventuelle des tags se déplace. Alors, la nouvelle zone est l'intersection des deux aires, soit B. On suit ces étapes jusqu'à ce que le robot ne détecte aucun tag.

### 5.5.2 Méthode probabiliste

Lorsque le robot se déplace, il vérifie s'il reçoit des tags. Si c'est vrai, il discrétise la zone selon le modèle de perception de l'antenne. Pour chaque particule, le robot vérifie si elle est reçue à partir des zones; sinon il est rejeté. Nous affectons une probabilité  $p(x|z_{1:t})$  à chaque particule.  $x$  est la position prédite du tag et  $z_{1:t}$  les observations entre les instants 1 et  $t$ . Nous utilisons la règle de Bayes (présentée dans l'annexe) qui considère l'indépendance des mesures consécutives. Soit la règle récursive de mise à jour :



## 5. CARTOGRAPHIE DES RADIO-ÉTIQUETTES

---

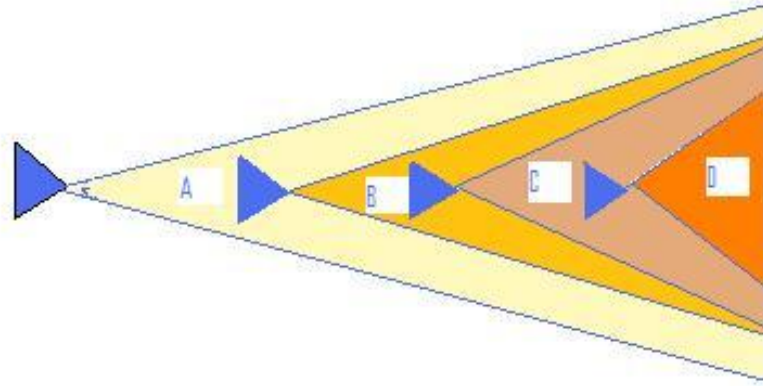


Figure 5.4: Mouvement du robot, à chaque étape, il réduit la zone estimée du tag

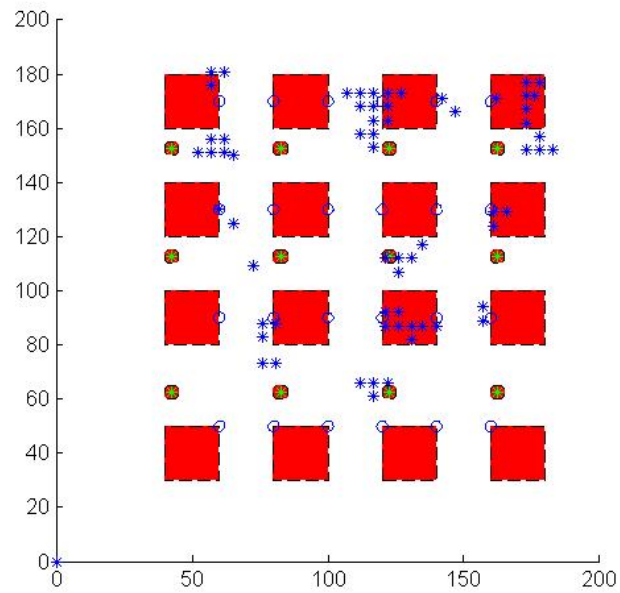


Figure 5.5: Positions estimées des tags représentées avec des étoiles bleues ( Premier algorithme)

## 5.5 Cartographie d'un environnement par des position de radio-étiquettes

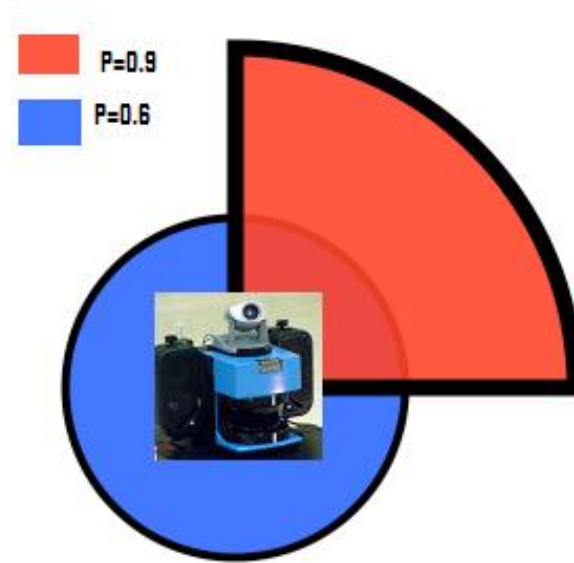


Figure 5.6: Un modèle de capteur simplifié de l'antenne du robot

$$p(x|z_{1:t}) = \alpha.p(z_t|x)p(x|z_{1:t-1})$$

Le principe de la méthode est représenté dans l'algorithme 7.

$p(z_t|x)$  spécifie la vraisemblance de l'observation considérant que le tag est dans la position  $x$ . Le modèle probabiliste de l'antenne est formé de deux composantes. La figure 5.6 montre l'intervalle de détection de chaque antenne. C'est un arc avec un angle d'ouverture de 95 degrés dans la direction de l'antenne. Les tags RFID qui sont proches de l'antenne sont toujours détectés. Cette zone est modélisée par un cercle autour du centre du récepteur. La figure 5.6 montre aussi la vraisemblance des deux intervalles de détections.

Nous évaluons notre méthode probabiliste en calculant pour les coordonnées  $x$  et  $y$  des tags, la différence entre la moyenne des positions estimées et les vraies positions. La figure 5.8 montre l'erreur sur  $x$  en bleu et sur  $y$  en vert. La précision est estimée à environ 0.2 m sur l'axe des  $x$  et 0.4 sur l'axe des  $y$ .

### 5.5.3 Discussion

La 5.5 montre une carte contruite par la méthode déterministe decrite par l'algorithme 7. Les vraies positions des tags sont au milieu de chaque arête et à l'extrémité de chaque rectangle (cercle bleu). Nous remarquons que les positions estimées des tags sont placées autour de

## 5. CARTOGRAPHIE DES RADIO-ÉTIQUETTES

---

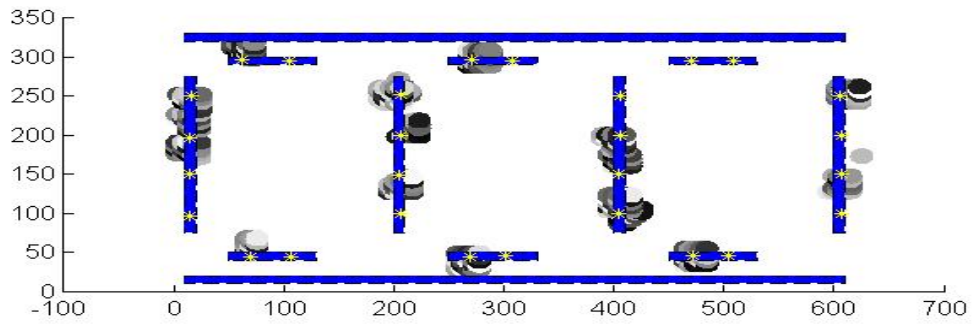


Figure 5.7: Les positions estimées des tags RFID. La couleur du cercle représente la probabilité a posteriori de la position correspondante

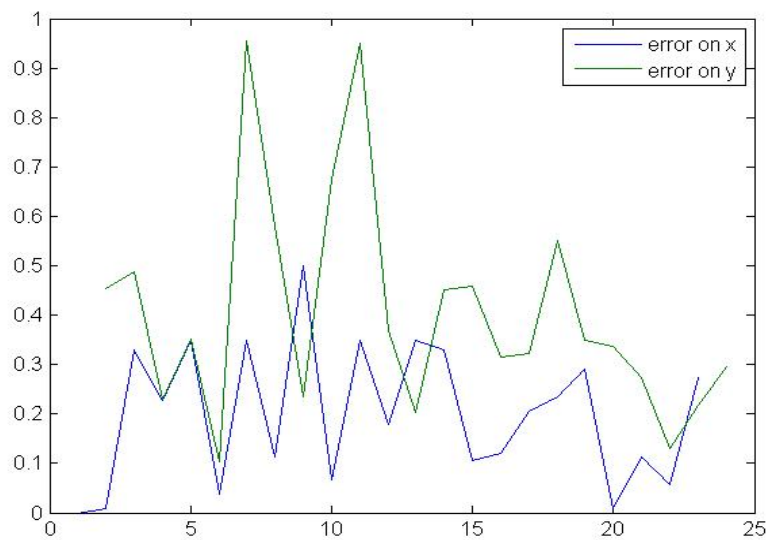


Figure 5.8: Erreur sur la position du tag

**Algorithm 7**


---

```

1: for tag  $\leftarrow$  1 to N do
2:   for robot-position  $\leftarrow$  1 to P do
3:     repeat
4:       R=Memorize the robot position
5:     until received-tags  $\neq \emptyset$ 
6:     P  $\leftarrow$  ellipse(robot-position)
7:     for  $x_i \leftarrow$  1 to size(P) do
8:

$$p(x_i|z_{1:t}) = \alpha.p(z_t|x_i)p(x_i|z_{1:t-1})$$

9:       if  $R_i$  receives  $p_i$  then
10:         reject  $p_i$ 
11:       end if
12:     end for
13:   end for
14: end for

```

---

chaque vraie position. 5.7 montre les positions estimées des tags calculées par la méthode probabiliste. Les disques en gris clair montre une faible probabilité et ceux avec un gris foncé sont de forte probabilité. Dans la figure 5.8, nous calculons la variance des coordonnées x et y. L'erreur maximale est de 0.9 pour l'axe des x et de 0.3 pour l'axe des y.

## 5.6 Conclusion

Dans ce chapitre, nous avons présenté un aperçu sur les méthodes de cartographie géométrique et topologique. La grille d'occupation est considérée comme très utile car elle gère les incertitudes sur les positions des tags avec les probabilités. Ensuite nous avons développé deux algorithmes de cartographie par les RFIDs.

- Une méthode déterministe qui utilise le modèle de perception des antennes RFID. Elle représente la position éventuelle d'un tag par un nuage de particules.
- Une méthode stochastique qui utilise un filtrage bayésien pour estimer la position d'un tag. Elle affecte à chaque particule un poids en utilisant un modèle probabiliste de capteur.

## 5. CARTOGRAPHIE DES RADIO-ÉTIQUETTES

---

# Conclusion générale

Au cours de ce travail nous avons étudié comment la localisation et la cartographie constituent une grande problématique en robotique.

Nous avons vu que plusieurs applications ont besoin de connaître le modèle spatial et la position courante du robot. Pour acquérir une carte précise de leurs environnements, les robots doivent répondre à un besoin majeur des véhicules mobiles autonomes. De plus, la plupart des approches de cartographie supposent que l'environnement du robot est statique et ne change pas dans le temps. Ce qui n'est pas réaliste puisque souvent les endroits dans lesquels évoluent les robots sont occupés par des humains.

Par ailleurs, la vision par ordinateur est une discipline qui a atteint sa maturité. L'énorme dissémination des caméras dans le grand public à travers les téléphones portables a accéléré la maîtrise de leur technologie et la baisse de leur coût. Cela leur permet de jouer un grand rôle en robotique. La vision permet en effet d'acquérir des données sensorielles sur l'environnement extérieur, données nécessaires pour faire naviguer le robot. Nous avons présenté des solutions à une série de problèmes dans ce contexte.

Dans le cadre du projet COMMROB, l'objectif principal est la communication de haut niveau avec et entre les robots. Ce modèle est comparable à celui de la communication humaine. Il met en jeu des comportements (comme les mouvements et les gestes) ainsi que leur reconnaissance pour l'interaction homme-robot.

Nous avons d'abord proposé une méthode d'indexation globale d'images qui permet de rechercher les images dans une base de données. Notre système utilise des attributs couleur et texture, obtenus en mettant en oeuvre une méthode pour fusionner des mesures de la couleur

## Conclusions générale

---

et de la texture. Cette méthode a été ensuite déclinée de manière locale: les attributs de couleur et texture permettent de caractériser le voisinage de points d'intérêt extraits par une approche innovante exploitant la couleur. La reconnaissance d'objets nécessite, en général, deux étapes : l'apprentissage et l'appariement. Nous avons développé un système de reconnaissance d'objets appliqué sur des images couleur texturées; ce système est fondé sur nos points d'intérêt, caractérisés par des descripteurs invariants à l'échelle et à l'orientation.

Ensuite, nous avons exploité nos travaux en Vision pour des applications Robotique. Nous avons utilisé des amers pour construire une carte de l'environnement dont la mise à jour se fait par appariement entre deux modalités sensorielles: des radio-étiquettes et des amers visuels ponctuels. Nous avons présenté un nouvel algorithme de localisation qui utilise des RFID. Ces derniers connaissent un essor considérable qui favorise leur utilisation. Nous utilisons une approche bayésienne fondée sur la méthode Monte-Carlo pour filtrer la position du robot et réduire donc l'incertitude. Nous avons aussi utilisé une méthode déterministe qui utilise le modèle de perception des antennes RFID pour recalibrer la position du robot à partir des observations des tags RFID. La précision a été améliorée à l'aide d'une localisation active.

Nous avons développé un algorithme de localisation et cartographie simultanée dans un environnement inconnu qui utilise des mesures des points d'intérêt extraits et caractérisés par notre approche depuis des images couleur texturées

Nous avons développé, enfin, une stratégie de navigation qui combine les données de capteurs provenant de radio-étiquettes RFID et d'une caméra monoculaire. De sorte que, si le robot ne reçoit pas de tags, il réduit l'incertitude avec les amers visuels extraits.

En matière de cartographie, l'approche probabiliste reste de loin la plus utilisée et la plus efficace. Nous avons cartographié notre environnement par des tags RFID en utilisant le filtrage bayésien qui permet de mettre à jour les positions des tags avec le mouvement du robot, et en supposant que durant cette phase, la localisation était donnée par les amers visuels préalablement appris.

# Perspectives

Nous nous intéressons à étendre le détecteur descripteur destiné à un objet pour la détection de classes d'objets en exploitant l'approche probabiliste développée par FeiFei ([FFFP03](#)).

Nous pouvons envisager aussi de développer une méthode de SLAM qui se base sur la technologie des RFID et qui utilise les grilles d'occupation pour construire les cartes de l'environnement. Ensuite, nous pouvons développer une méthode de SLAM qui fusionne à chaque instant les mesures des amers visuels et des RFID pour une navigation précise.

Par ailleurs, les techniques statistiques de la navigation qui réduisent l'incertitude sur la position du robot sont efficaces, certes; mais l'homme développe ces mêmes stratégies de navigation, de façon automatique et plus robuste. Nous envisageons donc développer une stratégie de navigation plus automatique. Elle sera basée sur les réseaux de neurones tels que les stimuli des champs de neurones donnés par les mesures d'amers visuels et des tags RFID ([OHP09](#)). Le robot pourrait ainsi déterminer sa trajectoire, ses obstacles et sa vitesse. Il aurait ainsi plus d'autonomie dans la gestion de ses moyens et plus d'efficacité dans les services qu'il rendrait à l'Homme.





# Publications

1. Y.Raoui, E.H. Bouyakhf, M. Devy, F. Regragui , Mapping and localization with RFID technology, Proc. 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2011), Noordwijkerhout (Netherlands), July 2011.
2. Y.Raoui, M.Devy, E.H.Bouyakhf, F. Regragui , Mobile Robot Localization Scheme based on Fusion of RFID and Visual Landmarks, Proc. 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2011), Noordwijkerhout (Netherlands), July 2011.
3. Y.Raoui, M.Goller, M.Devy, T.Kechner, J.M.Zollner, R.Dillmann, A.Cousto , RFID based topological and metrical self-localization in a structured environment, Proc. 14th International Conference on Advanced Robotics (ICAR 2009), Munich (Deutschland), June 2009.
4. Y. Raoui ,A. Ghazali, E. Hamzaoui, A. Mahmoudi, Y. Kabbadj, W. Ait Fares, F. Eddaoudi, F. Regragui, M.M. Himmi and E. Bouyakhf, Image and signal Processing and their Implication in Neurosciences, Medicine and Industrial Applications, Proc. Humboldt-Kolleg University Research-Industry: partnering opportunities, Rabat (Morocco), June 2010, .
5. Y. Raoui, E.H.Bouyakhf, M. Devy, Unified Color Texture Descriptor Using Gaussian Color Model and Rayleigh Density Functions, SETIT 2007, Proc. International Conference on Siences of Electronics, Technologies of Information and Telecommunications (SETIT 2007), Hammamet (Tunisia), March 2007.
6. Y.Raoui, M.Devy, E.H.Bouyakhf , Scale and Rotation Invariant Descriptor Using Harris Laplacian Decriptorin Color Textured Images, Proc. 5th International Conference on

## Publications

---

- Sciences of Electronics, Technologies of Information and Telecommunications (SETIT 2007), Hammamet (Tunisie), March 2009.
7. Y.Raoui, M.Devy, E.H.Bouyakhf, Indexation d'images couleur texture: extraction de points d'interets par le detecteur de harris laplace, Proc. Congrès des doctorants de l'Ecole Doctorale Systèmes (EDSYS 2009), Toulouse (France), May 2009.
  8. Y.Raoui, M.Devy, E.H.Bouyakhf and F. Regragui, Global and Local Image Descriptors for Content Based Image Retrieval and Object Recognition, in Applied Mathematical Sciences, Vol. 5, no. 42, p. 2109 - 2136, 2011.
  9. Y.Raoui, M.Devy, E.H.Bouyakhf and F. Regragui, Metrical Mapping and Self localization with RFID Technology and Visual Landmarks, in Applied Mathematical Sciences, Vol. 5, no. 42, p. 2137 - 2157, 2011.
  10. Y.Raoui, M.Devy, E.H.Bouyakhf, Image Indexing for Object Recognition and Content Based Image Retrieval. in International Journal of Research and Reviews in Computer Science, Vol 1, No 4, December 2010.

# Annexe: Filtrage Probabiliste

La robotique probabiliste est une nouvelle approche de robotique qui s'intéresse aux incertitudes dans la perception et l'action du robot. En effet, les probabilités sont capables de traiter les différentes incertitudes qui caractérisent normalement les mesures acquises ou les actions effectuées par un robot mobile. Dans notre travail, nous utilisons l'approche probabiliste dans plusieurs méthodes que nous développons. Nous présentons ici les règles utilisées pour traiter les incertitudes qui sont dues à plusieurs causes :

- Incompatibilité des connaissances sur l'environnement ;
- Mesures bruitées et imprécision pouvant agir sur l'ensemble du système ;
- Défaillances, de toutes formes, ainsi que les erreurs algorithmiques pouvant générer des anomalies dans le fonctionnement du robot

## Probabilité Conditionnelle et Loi de Bayes

Soient,  $x$  et  $y$  deux événements :

### La probabilité d'une disjonction

$$P(x \cup y) = P(x) + P(y) + P(x \cap y)$$

Dans le cas où  $x$  et  $y$  sont indépendants, cette dernière règle s'écrit comme suit:

$$P(x \cup y) = P(x) + P(y)$$

### La probabilité conditionnelle

$$P(x|y) = \frac{P(x \cup y)}{P(y)}$$

D'où:  $P(x \cap y) = P(x|y)P(y) = P(y|x)P(x)$

Le terme  $P(x)$  est la probabilité de  $x$  a priori. Le terme  $P(x|y)$  est appelé la probabilité a posteriori de  $x$  sous condition de  $y$

Pour des événements indépendants :

$$P(x|y) = P(x) \Rightarrow P(x \cap y) = P(x)P(y)$$

En plus :  $P(\cap_{i=0}^n x_i) = P(x_0)P(x_1|x_0)P(x_2|x_1 \cap x_0) \dots P(x_n | \cup_{i=0}^{n-1} x_i)$

Par ailleurs,

On appelle distribution marginale de  $y$  par rapport à  $x$  , la distribution de probabilité suivante :

$$P(y) = \sum_x P(y|x)P(x)$$

Le théorème de Bayes est utilisé pour déterminer ou mettre à jour l'estimation d'une probabilité d'un événement à partir d'une ou de plusieurs observations. Ce théorème énonce en conséquence une probabilité conditionnelle pour évaluer la valeur de  $x$  en sachant la valeur de  $y$  si elle est connue :

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

Le terme  $P(y|x)$  pour un  $y$  connu est appelé la fonction de vraisemblance de  $x$  . De même, le terme  $P(y) \neq 0$  est appelé probabilité a priori de  $y$  , qui peut être remplacé par la probabilité marginale, pour nous donner

$$P(x|y) = \frac{P(y|x)P(x)}{\sum_x P(y|x)P(x)}$$

Le dénominateur exprime toutes les réalisations possibles de  $y$  et peut être considéré comme une constante de normalisation  $\nu$ .

$$P(x|y) = \nu P(y|x)P(x)$$

### L'hypothèse de Markov

L'hypothèse de Markov propose que l'état présent ne dépende que d'un nombre fini d'états passés. Ce qui s'exprime comme suit :

$$P(x_t|x_{0:t-1}) = P(x_t|x_{t-1}, x_{t-2}, \dots, x_{t-n})$$

Au premier ordre, cette hypothèse peut se traduire par la formulation suivante où l'état présent ne dépend que de l'état précédent :

$$P(x_t|x_{0:t-1}) = P(x_t|x_{t-1})$$

Cette hypothèse permet en particulier de réduire la complexité et le temps du calcul.

### Le Filtre Bayésien

Le filtre bayésien est un estimateur récurrent d'état. Dans le cas gaussien, la probabilité de l'état a posteriori est représentée comme une distribution normale autour de l'état vrai décrite par l'équation suivante :

$$P(x) = \det(2\pi C)^{\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \xi)^T C^{-1}(x - \xi)\right)$$

$x$ : est un vecteur d'état,  $\xi$  est un vecteur de moyenne (qui a une dimension égale à celle de l'état),

$C$  est une matrice de covariance symétrique et semi positive.

Le filtre de Bayes représente l'algorithme le plus général pour calculer une connaissance du robot de l'état de son environnement à partir de différentes mesures et commandes; par exemple, déterminer en ligne la localisation du robot dans son environnement en utilisant ses propres moyens (ses capteurs). Cette connaissance se représente par l'ensemble des distributions conditionnelles probabilistes sur les données disponibles. Chaque densité probabiliste décrit une hypothèse possible dans l'espace d'état.

$$p(x_t) = p(x_t|z_{1:t}, u_{1:t})$$

$x_t$  représente l'état à un instant donné ( $t$ ),  $(z_{1:t}, u_{1:t})$  sont respectivement l'ensemble des mesures et des commandes sur tous les instants passés jusqu'à l'instant courant.

L'algorithme du filtre de Bayes respecte l'hypothèse de Markov et se compose de deux étapes fondamentales qui se répètent:

1. L'étape 1 de prédiction :

$$\hat{p}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})$$

$p(x_{t-1})$  décrit une probabilité de l'état  $x_{t-1}$  a priori.

$p(x_t|u_t, x_{t-1})$  représente la probabilité pour qu'une cause (ici une commande :  $u_t$ ) qui peut arriver à l'instant  $t$ , soit capable de provoquer une transition d'état.

2. L'étape 2 de correction :

$$p(x_t|z_t) = \nu p(z_t|x_t)\hat{p}(x_t)$$

$p(z_t|x_t)$  décrit la probabilité de recevoir une mesure  $z_t$  en sachant que l'état courant  $x_t$  et  $\hat{p}(x_t)$  résulte de la première étape.

## Filtre de Kalman

Le filtre de Kalman est un filtre récursif capable d'estimer l'état d'un système dynamique à partir d'observations sur le système, observations qui peuvent être bruitées. Pour que cette estimation aie un sens, il est nécessaire que le système dynamique modélisé soit linéaire, c'est à dire que l'état courant s'obtienne par une évolution linéaire de l'état précédent.

Quand on parle du caractère récursif du filtre, on parle du fait que l'état courant peut être estimé à partir de l'état précédent et des mesures que l'on fait. En effet, le filtre de Kalman a à la base une chaîne de Markov (sur laquelle on a fait agir des opérateurs linéaires éventuellement perturbés par un bruit gaussien).

Le modèle du système dynamique considéré comme valide a priori quand on applique un filtrage de Kalman est le suivant :

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

où :

- $x$  représente l'état
- $k$  représente le moment courant
- $F_k$  décrit la manière de passage de l'état au moment  $k - 1$  à l'état au moment  $k$
- $B_k$  décrit la manière dont les entrées influent sur l'état
- $z_k$  représente les observations que l'on a sur le système

- $H_k$  décrit le modèle d'observation et
- $w_k$  et  $v_k$  sont des bruits gaussiens de moyenne nulle

Ces deux derniers bruits sont les sources d'erreur dont on cherche à diminuer l'importance.

Le filtre étant récursif, on peut passer d'une étape à l'autre de l'estimation sans prendre en compte des étapes plus anciennes. Pour chaque étape, le filtre est caractérisé par seulement deux paramètres : l'état estimé et l'imprécision que l'on pense avoir sur cette estimation. Soient les notations suivantes :

- $\bar{x}_{k|k}$ , l'état au moment k estimé à partir des observations jusqu'au moment k (inclusif)
- $\bar{x}_{k|k-1}$ , l'état au moment k estimé à partir des observations jusqu'au moment k (exclusif)
- $P_{k|k}$ , la matrice de covariance, mesure de l'erreur sur l'estimation de l'état, au moment k et prenant en compte les observations jusqu'au moment k (inclusif)
- $P_{k|k-1}$ , la matrice de covariance, mesure de l'erreur sur l'estimation de l'état, au moment k et prenant en compte les observations jusqu'au moment k-1 (exclusif)

Le passage d'une étape d'estimation à la suivante se fait en deux pas. D'abord on fait la prédiction :

$$\begin{aligned}\bar{x}_{k|k-1} &= F_k \bar{x}_{k-1|k-1} + B_{k-1} u_{k-1} \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + \text{Variance}(w_k)\end{aligned}$$

Ensuite, on fait une correction de l'état ainsi obtenu en regardant la différence entre l'observation perçue et celle attendue de notre état prédit :

$$\begin{aligned}\tilde{y}_k &= z_k - H_k \bar{x}_{k|k-1} \\ S_k &= H_k P_{k|k-1} H_k^T + \text{Variance}(v_k) \\ K_k &= P_{k|k-1} H_k^T S_k^{-1}\end{aligned}$$

$$\bar{x}_{k|k} = \bar{x}_{k|k-1} + K_k \tilde{y}_k$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$



où:

- $\tilde{y}_k$  représente l'observation « résiduelle »
- $S_k$  représente la covariance « résiduelle »
- $K_k$  représente le gain optimal de Kalman

L'idée est de trouver le coefficient qui permet de passer de la différence entre l'observation réelle et celle estimée à la différence, entre l'état réel et l'état estimé. Le gain de Kalman optimal est un tel coefficient et on prouve que, dans le cas où les hypothèses que l'on avait fait sur la structure du système et sur l'état initial sont respectées, on a les conclusions suivantes :

$$Esperance(x_k - \bar{x}_{k|k}) = Esperance(x_k - \bar{x}_{k|k-1}) = 0$$

$$P_{k|k} = Covariance(x_k - \bar{x}_{k|k})$$

$$P_{k|k-1} = Covariance(x_k - \bar{x}_{k|k-1})$$

## Filtrage particulière - Méthode de Monte Carlo

### Introduction aux méthodes de localisation Markoviennes

La localisation par la méthode de Monte Carlo est un algorithme probabiliste faisant partie de la catégorie plus large des méthodes Markoviennes. L'idée qui est à la base de cette famille d'algorithmes est d'avoir à chaque moment une distribution de probabilité sur les positions possibles du robot. Distribution que l'on met à jour à chaque fois que l'on a un déplacement ou une observation (parvenue des capteurs). Le passage d'une distribution à l'autre se fait par convolution et propagation bayésienne. On va détailler dans ce qui suit les mécanismes de propagation. Pour intégrer un mouvement du robot, un modèle a priori défini du déplacement est nécessaire. Ce modèle doit fournir la probabilité  $P(pos_{suivante} | pos_{courante}, déplacement)$  de se trouver à une position suivante étant données la position courante et le déplacement enregistré. Le fait que le modèle ne soit pas déterministe mais probabiliste, permet de prendre en compte la présence d'erreurs dans le modèle du robot, c'est-à-dire dans notre cas, de l'odométrie. Étant donné ce modèle, on peut propager la connaissance sur la position suite à un déplacement de la façon suivante, habituelle aux modèles markoviens :

$$C(position) = \int (P(pos_{suivante} | pos_{courante}, déplacement) C(position_{precedente}) d[position_{precedente}]$$

Avec C représente la croyance.

Pour intégrer une observation enregistrée par les capteurs on amplifie la distribution de probabilité par une mesure bayésienne de la concordance de chaque position avec l'observation :

$$C(position) = K.\omega(observation|position).C(position)$$

où  $w$  est le poids, et  $K$  est choisi de telle façon que la distribution de probabilité  $C$  reste normalisée à 1. Une telle façon d'évoluer (n'est) est pertinente que dans le cas où les hypothèses markoviennes sont satisfaites, c'est-à-dire quand les réponses courantes des capteurs sont indépendantes des réponses précédentes, cas dans lequel on va supposer que l'on se trouve.

L'avantage de cette approche est sa généralité. Ainsi, on n'a pas a priori des limitations sur la complexité de la distribution de probabilité de l'état courant et ceci servira énormément dans certains cas. Le plus significatif étant celui où le robot s'est «perdu» ou bien, quand il démarre sans avoir de connaissances sur sa position initiale : dans ces cas, l'approche du filtre de Kalman s'avère impuissante.

## Spécificités de la méthode de Monte Carlo

La méthode de Monte Carlo se distingue des autres algorithmes markoviens par la façon dont la connaissance sur la position est représentée. Pour décrire la distribution de probabilité de la position on utilise un ensemble de «particules», chacune représentant une position précise et ayant associé une probabilité. Par cette façon de modéliser la connaissance sur l'état courant, cette méthode s'inscrit dans la catégorie des algorithmes appelés filtres particulaires. Comme on va le voir, cette méthode garde en grande partie les avantages des approches markoviennes génériques (représentation multimodale de la connaissance sur l'état courant) mais réussit aussi, en approximant une distribution a priori continue par une distribution discrète, de s'encadrer dans des limites de temps de calcul et d'occupation de la mémoire très convenables.

L'algorithme 5 est itératif et chaque pas  $k$  de l'itération comprend 3 étapes distinctes :

1. l'étape de prédiction de mouvement :

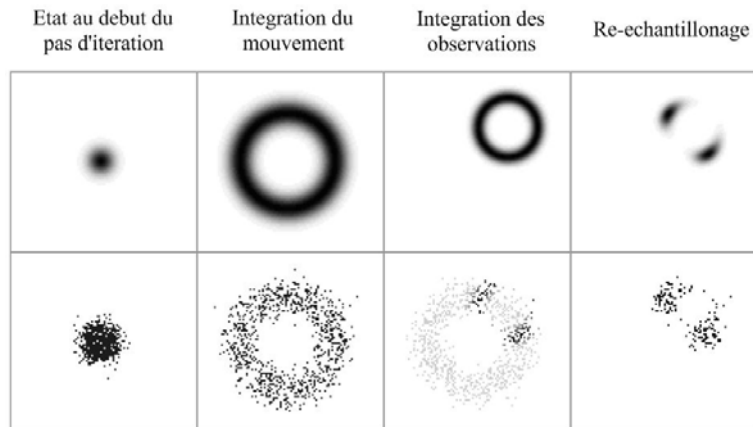


Figure 5.9: Étapes de chaque pas d'itération de la méthode de Monte Carlo

Cette étape a comme entrée l'ensemble précédent de particules  $F_{k-1}$  ainsi que le déplacement  $u_k$  enregistré depuis l'étape  $k-1$ . Elle utilise aussi le modèle  $P$  que l'on a sur le mouvement. Pour chaque particule de l'ensemble  $F_{k-1}$  on fait un tirage aléatoire uniforme d'une position suivante possible avec les probabilités données par  $P(F_{k-1}[i], u_k)$ . Ainsi on obtient l'ensemble  $M_k$ , sans oublier d'associer à chaque particule  $M_k[i]$  la probabilité correspondant avant à  $F_{k-1}[i]$ .

## 2. l'étape d'incorporation des observations :

Cette étape prend en entrée l'ensemble de particules  $M_k$  ainsi que toutes les observations  $w_k$  enregistrées depuis l'étape  $k-1$ . Elle utilise aussi la mesure probabiliste  $\omega$  que l'on a sur la concordance entre l'état courant et les observations.

En amplifiant le poids probabiliste dans l'ensemble  $M_k$  de chaque particule par sa concordance avec les observations, et puis en renormalisant, on trouve l'ensemble  $O_k$ .

Cette étape prend en entrée l'ensemble  $O_k$  et génère l'ensemble  $F_k$  en tirant au sort (avec répétition) uniformément  $|O_k|$  particules parmi les particules dans  $O_k$  proportionnellement aux probabilités calculées au pas précédent. Les probabilités associées à ces nouvelles particules seront toutes égales entre elles donc égales à  $\frac{1}{|O_k|}$ .

## 3. Comme tout algorithme itératif, cette méthode a besoin en plus de ceci d'une étape d'initialisation, donc d'un «pas zéro». Dans notre cas cette étape dépend beaucoup du problème que l'on cherche à résoudre. Plus précisément, si l'on veut juste maintenir une bonne estimation de la position du robot («motion tracking»), a priori on sait d'où on part, donc on initialise le filtre particulaire avec une distribution centrée sur de la

**Algorithm 8** Filtrage particulaire

---

```

1: for  $i \leftarrow 1$  to  $|M_k|$  do
2:    $O_k[i].position \leftarrow M_k[i].position$ 
3:    $O_k[i].probabilite \leftarrow M_k[i].probabilite$ 
4:   for  $j \leftarrow 1$  to  $|w_k|$  do
5:      $O_k[i].probabilite \leftarrow O_k[i].probabilite \cdot \omega(w_k[j] | O_k[i].position)$ 
6:   end for
7: end for
8: for  $i \leftarrow 1$  to  $|M_k|$  do
9:    $O_k[i].probabilite \leftarrow O_k[i].probabilite / \text{sum}(O_k[:].probabilite)$ 
10: end for

```

---

position estimée de départ. Si par contre on veut localiser le robot, a priori on n'a pas beaucoup d'informations sur sa position initiale, alors on va initialiser le filtre avec une distribution uniforme de positions de l'environnement. Au fur et à mesure que le robot avance et reçoit des observations, il y a une grande probabilité que les fausses positions soient éliminées et que l'on reste avec un ensemble d'échantillons qui donne une bonne estimation de la position réelle.



# Bibliography

- [ATSB04] I. Ashokaraj, A. Tsourdos, P. Silson, and B.White. Sensor based robot localisation and navigation: Using interval analysis and extended kalman filter. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004. 90
- [Bal81] D.H Ballard. Generalizing the hough transform to detect arbitrary patterns. *Pattern Recognition*, 13:111–122, 1981. 67
- [BM00] R. Brunelli and O. Mich. Image retrieval by examples. *IEEE Trans. on Multimedia*, 2000. 25
- [BMP01] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. Int. Conf. Computer Vision (ICCV)*, 2001. 42
- [BNL<sup>+</sup>03] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003. 107
- [Bry09] M. Bryson. Active and action based slam. In *Proc. SLAM Summer School*, 2009. 96
- [BTG06] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. 9th European Conf. on Computer Vision (ECCV), Graz, Autriche., 2006.* 42, 43
- [BTM03] S. Bhagavathy, J. Tesic, and B.S. Manjunath. On the rayleigh nature of gabor filter outputs. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2003. 33

## BIBLIOGRAPHY

---

- [BZ99] G. Bianco and A. Zelinsky. Biologically inspired visual landmark learning and navigation for mobile robots. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1999. 105
- [CH00] P. Chang and M. Hebert. Omni-directional structure from motion. In *Proc. IEEE Int. Workshop on Omnidirectional Vision, Hilton Head Island (USA)*, 2000. 104, 105
- [CH02] O. Carmichael and M. Hebert. Object recognition by a cascade of edge probes. In *Proc. British Machine Vision Conf. (BMVC)*, 2002. 42
- [CL85] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), St Louis (USA)*, 1985. 103
- [CO96] C. Carson and V.E. Ogle. Storage and retrieval of feature data for a very large online image collection. *IEEE Computer Society Bulletin of the Technical Committee on Data Engineering*, 1996. 25
- [Dau80] J.G. Daugman. Two-dimensional spectral analysis of cortical receptive field profile. *Vision Research*, 20:847 – 856, 1980. 28
- [Dau85] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal Optical Society of America (JOSA) A*, 2(7):1160 – 1169, 1985. 66
- [Dav98] A.J. Davison. Mobile robot navigation using active vision. In *PhD dissertation, Univ. of Oxford*,, 1998. 107, 110
- [DDBC00] C. Drocourt, L. Delahoche, E. Brassart, and C. Cauchois. Experimental result got with omnidirectional vision sensor. In *Proc. IEEE Workshop on Omnidirectional Vision, Hilton Head Island (USA)*, 2000. 106
- [DFBT99] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1999. 75
- [DM98] A.J. Davison and D.W. Murray. Mobile robot localisation using active vision. In *Proc. Fifth European Conf. Computer Vision (ECCV)*, pages 809 – 825, 1998. 107

- 
- [DM02] A.J. Davison and D.W. Murray. Simultaneous localization and map-building using active vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24(7):865 – 880, July 2002. 107
- [EIP97] S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. In *Unpublished manuscript: <http://kybele.psych.cornell.edu/~edelman/archive.html>*, 1997. 61
- [Elf92] A. Elfes. Dynamic control of robot perception using multi-property inference grids. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1992. 124
- [FA91] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, pages 891 – 906, 1991. 56
- [FFFP03] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2003. 133
- [Fil06] D. Filliat. Robotique mobile. In *Support de cours donn  e    l'ENSTA, Paris*, 2006. 15
- [FW09] N. Fairfield and D. Wettergreen. Active slam and loop prediction with the segmented map using simplified models. In *Proc. Int. Conf. on Field and Service Robotics (FSR)*, 2009. 96
- [GdBSG01] J.M. Geusebroek, R. Van den Boomgaard, A. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(12):1338–1350, 2001. 17, 25, 64
- [Haw70] J.K. Hawkins. *Texture Properties for Pattern Recognition*. in *Picture Processing and Psychopictorics*, B.C. Lipkin and A.Rosenfeld, Eds, Academic Press, New York, 1970. 24
- [HBF<sup>+</sup>05] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Filipose. Mapping and localization with rfid technology. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005. 81
- [HGS05] M.A. Hoang, J.M. Geusebroek, and A.W.M. Smeulders. Color texture measurement and segmentation. *Journal Signal Processing - Special section on content-based image and video retrieval*, 85(2), February 2005. 43



## BIBLIOGRAPHY

---

- [HSH93] K. Hirata, Y. Hara, N. Shibata, and F. Hirabayashi. Media-based navigation for hypermedia systems. In *Proc. 5th ACM conference on Hypertext*, 1993. 24
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, 1988. 42, 43, 48, 59
- [JL03] I. Jung and S. Lacroix. High resolution terrain mapping using low altitude aerial stereo imagery. In *Proc. 9th Int. Conf. on Computer Vision (ICCV)*, 2003. 107
- [JS04] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004. 43
- [KB01] T. Kadir and J.M. Brady. Scale, saliency and scene description. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2001. 43, 54
- [KCS07] A. Kleiner, C.Dornhege, and S.Dali. Mapping disaster areas jointly: Rfid-coordinated slam by humans and robots. In *Proc. IEEE Int. Workshop on Safety, Security and Rescue Robotics (SSRR), Rome, Italy*, 2007. 81
- [Koe84] J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5), 1984. 44
- [KS03] J.H. Kim and S. Sukkarieh. Airborne simultaneous localisation and map building. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003. 107
- [KSP<sup>+</sup>03] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer. Distributed search and rescue with robot and sensor team. In *Proc. 4th Int. Conf. on Field and Service Robotics (FSR)*, 2003. 81
- [KvD87] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. In *Biological Cybernetics*, 1987. 45
- [Lin94] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994. 43, 44, 52
- [Low99] D.G. Lowe. Object recognition from local scale invariant features. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 1999. 42, 43, 67
- [Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision (IJCV)*, 2004. vi, 17, 57, 58, 61

- [MGD98] P. Montesinos, V. Gouet, and R. Deriche. Differential invariants for color images. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, 1998. 63
- [MM96] B.S. Manjunath and W.S. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 18:837 – 842, 1996. 34
- [MN95] H. Murase and S. Nayar. Visual learning and recognition. *Int. Journal of Computer Vision (IJCV)*, 1995. 42
- [MS04] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int. Journal on Computer Vision (IJCV)*, 2004. 17, 42, 44, 50, 53, 70
- [MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10), October 2005. 43
- [MTKW02] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proc. National Conf. on Artificial Intelligence (AAAI)*, 2002. 81
- [NRT97] J. Neira, M.I. Ribeiro, and J.D. Tardos. Mobile robot localisation and map building using monocular vision. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Open Systems (IROS)*, 1997. 107
- [OHP09] M. Oubbati, W. Holoch, and G. Palm. Neural fields for complex behavior generation on autonomous robots. In *Proc. Int. Joint Conf. on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009*. 133
- [OM95] V.E. Ogle and M. Stonebraker. Chabot: Retrieval from a relational database of images. In *IEEE Computer*, 1995. 25
- [PA02] A. Paulino and H. Araujo. Pose estimation for central catadioptric systems : an analytical approach. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, 2002. 106
- [Pic70] M. Pickett. *Visual Analysis of the texture in the Detection and Recognition of Objects*. in *Picture Processing and Psychopictorics*, B.C. Lipkin and A. Rosenfeld, Eds, Academic Press, New York, 1970. 24

## BIBLIOGRAPHY

---

- [Pon] Data set du groupe de recherche de J. Ponce, Urbana Champaign (USA). [http://www-cvr.ai.uiuc.edu/ponce\\_grp/data/](http://www-cvr.ai.uiuc.edu/ponce_grp/data/). 70
- [Pon04] J. Ponce. Towards 3d object recognition. In *Congrès de Reconnaissance des Formes et Intelligence Artificielle, Toulouse, France*, 2004. 43
- [RBD07] Y. Raoui, E.H. Bouyakhf, and M. Devy. Unified color texture descriptor using gaussian color model and rayleigh density function. In *Proc. Int. Conf. on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet (Tunisia)*, 2007. 21
- [RGD<sup>+</sup>09] Y. Raoui, M. Goller, M. Devy, T. Kerscher, J. Zollner, R. Dillmann, and A. Coustou. Rfid-based topological and metrical self-localization in a structured environment,. In *Proc. Int. Conf. on Advanced Robotics (ICAR), Munich (Germany)*, 2009. 101
- [SEGL05] R. Sim, P. Elinas, M. Griffin, and J.J. Little. Vision-based slam using the rao-blackwellised pconference filter. In *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2005. 107
- [SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(5), 1997. 60
- [SPH98] T. Svoboda, T. Pajdla, and V. Hlavac. Motion estimation using central panoramic cameras. In *Proc. IEEE Conf. on Intelligent Vehicles (IV)*, 1998. 105
- [SSC86] R.C. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *2nd Workshop on Uncertainty in Artificial Intelligence (AAAI), Philadelphia (USA)*, August 1986. 103
- [Svo99] T. Svoboda. Central panoramic cameras design, geometry. In *PhD, Cezch Technical University, Karlovo namesti 13, 12135 Prague 2, Cezch Republic*, September 1999. 105
- [SVS03] O. Shakernia, R. Vidal, and S. Sastry. Multibody motion estimation and segmentation from multiple centarl panoramic views. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003. 105
- [TFB05] S. Thrun, D. Fox, and W. Burgard. *Probabilistic Robotics*. 2005. 102

- [TMZ00] S. Thompson, T. Matsui, and A. Zelinsky. Localisation using automatically selected landmarks from panoramic images. In *Proc. Australian Conf. on Robotics and Automation, Melbourne (Australia)*, 2000. 105
- [TP91] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Maui, Hawaii (USA)*, June 1991.
- [Tsu03] T. Tsukiyama. Navigation system for mobile robots using rfid tags. In *Proc. Int. Conf. on Advanced Robotics (ICAR)*, 2003. 82
- [VSYZ08] P. Vorst, S. Schneegans, B. Yang, and A. Zell. Self-localization with rfid snapshots in densely tagged environments. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008. 81
- [VVS02] R. Frizera Vassallo, J. Santos Victor, and H.J. Schneebeli. A general approach for egomotion estimation with omnidirectional images. In *Proc. IEEE Int. Workshop on Omnidirectional Vision, Copenhagen (Denemark)*, 2002. 105
- [Wik] Wikipédia. 22
- [Wit83] A.P. Witkin. Scale-space filtering. In *Proc. 8th Int. Joint Conf. on Artificial Intelligence (IJCAI), Karlsruhe, Germany*, pages 1019 – 1022, 1983. 44
- [YTH<sup>+</sup>04] K. Yamano, K. Tanaka, M. Hirayama, E. Kondo, Y. Kimuro, and M. Matsumoto. Self-localization of mobile robots with rfid system by using support vector machine. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004. 82
- [ZLH07] Y. Zhou, W. Lu, and P. Huang. Laser-activated rfid-based indoor localization system for mobile robots. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pages 4600 -4605, Rome, Italy*, 2007. 82

## BIBLIOGRAPHY

---