

Année 2004

THESE

Préparé au
Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

En vue de l'obtention du titre de
Docteur de l'Institut National Polytechnique de Toulouse

Spécialité : **Informatique et Télécommunications**

par
Antonio MARIN HERNANDEZ

Vision dynamique pour la navigation d'un robot mobile

Soutenue le 9 février 2004 devant le jury :

Président **R. CHATILA**

Directeur de thèse **M. DEVY**

Rapporteurs **A. CASALS**
F. JURIE

Examineurs **A. AYACHE**

A Andréa et a mes parents

Remerciements

Avant tout je tiens à remercier MM. Jean-Claude Laprie et Malik Ghallab directeurs successifs du LAAS-CNRS, qui m'ont accueilli dans ce Laboratoire.

Je tiens également à remercier M. Raja Chatila responsable du groupe de recherche Robotique et Intelligence Artificielle (RIA) du LAAS-CNRS, pour m'avoir permis de travailler dans ce groupe pour réaliser cette thèse.

Je remercie Alicia Casals, professeur à l'Universitat Politecnica de Catalunya à Barcelone, et Frédéric Jurie, chargé de recherche CNRS au LASMEA à Clermont-Ferrand, pour avoir accepté d'être rapporteur sur ma thèse, ainsi que M. Alain Ayache pour avoir accepté de participer à mon jury de thèse.

Un grand remerciement à Michel Devy, mon directeur de thèse pour son soutien tout au long de la préparation de cette thèse ainsi que pour sa patience à l'heure de corriger le manuscrit.

Je voudrais remercier les membres du groupe RIA, permanents et doctorants, qui ont fait de mon séjour en France et plus précisément dans le groupe RIA, une expérience inoubliable. Je voudrais remercier très spécialement Sara Fleury pour toute l'aide donnée pour réaliser la *manip*. Enfin je voudrais remercier tous les membres du collectif *mexlaas*, pour m'avoir supporté pendant tout ce temps.

Table de matières

THÈSE.....	i
Table de matières	vii
Table des figures	xi
Introduction générale.....	1
Vision Dynamique et Navigation d'un robot mobile	5
1.1 Objectif de nos travaux	5
1.2 Navigation en milieu intérieur.....	8
1.2.1 Navigation métrique.....	9
1.2.2 Navigation avec une carte topologique.....	10
1.3 Navigation en extérieur.....	12
1.4 Suivi d'objets : Etat de l'art.....	13
1.4.1 Méthodes de suivi	14
1.4.2 Contrôle d'exécution du suivi visuel	15
1.5 Vision Active.....	17
1.6 Architecture du système - l'architecture LAAS.....	18
1.7 Conclusions	21
Chapitre 2 Méthodes de suivi visuel	23
2.1 Introduction	23
2.2 Snakes ou Contours Actifs.....	24
2.2.1 Formulation classique de l'énergie.....	24
2.2.2 Définition des modèles déformables.....	27

2.2.3	Snakes Couleur	32
2.3	Suivi avec corrélation 1D.....	36
2.4	Ensemble des points.....	39
2.5	Suivi par différences des Motifs.....	41
2.5.1	Approche de Hager et Belhumeur.	41
2.5.2	Approche de Jurie et Dhome	44
2.5.3	Calcul de la Matrice d'Interaction	50
2.6	Conclusions	52
Chapitre 3	Caractérisation de Méthodes de Suivi.....	53
3.1	Caractérisation des méthodes de suivi.....	53
3.1.1	Contours Actifs.....	55
3.1.2	Corrélation 1D	56
3.1.3	Ensemble de points.	57
3.1.4	Différences de Modèles.....	58
3.1.5	Initialisation des méthodes de suivi.....	59
3.2	Détection et reprise d'erreurs	61
3.2.1	Contours Actifs.....	61
3.2.2	Corrélation 1D	63
3.2.3	Ensemble de points	64
3.2.4	Différences des Modèles	65
3.3	Comparaison des méthodes de suivi.....	66
3.3.1	Fréquence.	66
3.3.2	Cibles et environnements	67
3.3.3	Caractéristiques des méthodes et détection d'erreurs.	69
3.3.4	Avantages et inconvénients	70
3.4	Analyse de performances pour différents contextes	70
3.4.1	Caractérisation des contextes pour l'évaluation des performances.....	72

3.4.2	Caractérisation de cibles et d'environnements.....	73
3.4.3	Déformations et variations de forme de la cible.....	78
3.5	Conclusions	80
Chapitre 4 Planification Perceptuel et Collaboration entre méthodes de suivi		83
4.1	Introduction	83
4.2	Planification perceptuel.....	84
4.2.1	Approche initiale de planification pour la navigation métrique	85
4.2.2	Visibilité des amer plans	91
4.2.3	Critères d'utilité.....	94
4.3	Enchaînement et permutation entre méthodes de suivi	104
4.3.1	Transition entre méthodes dans la présence des erreurs	105
4.3.2	Changements des cibles	107
4.3.3	Autres modules : contrôleur du module de suivi, observateurs....	109
4.4	Conclusions	110
Chapitre 5 Intégration et expérimentation		111
5.1	Introduction	111
5.2	Commande des paramètres de la camera	112
5.2.1	Angle du champ visuel - Zoom	113
5.2.2	Angles azimut et site.....	115
5.2.3	Résultats du contrôle d'une caméra active.....	116
5.3	Simulation (environnement virtuel d'intérieur)	116
5.4	Intégration et résultats expérimentaux.....	119
5.4.1	Intérieur – navigation par suivi d'amers	120
5.4.2	Intérieur - navigation par suivi de primitives	121
5.4.3	Navigation en extérieur pour le suivi de routes ou chemins.	124
5.4.4	Suivi d'obstacles en milieu routier.	126
5.5	Conclusions	127

Conclusion générale et perspective	129
Références Bibliographiques	133

Table des figures

<u>Figure 1.1. Schéma simplifié de tâches à l'intérieur du problème de la navigation autonome.</u>	7
<u>Figure 1.2. Différents environnements pris en compte pour la navigation a) en milieu intérieur et b) en extérieur.</u>	8
<u>Figure 1.3. Exemple d'un modèle géométrique du monde.</u>	9
<u>Figure 1.4. Schéma général sur le suivi.</u>	14
<u>Figure 1.5. L'architecture LAAS: vue d'ensemble.</u>	19
<u>Figure 1.6. Robots démonstrateurs du groupe RIA : a) Diligent, b) Dala.</u>	20
<u>Figure 1.7. Architecture générique du système visuel.</u>	21
<u>Figure 2.1. Potentiel externe défini à partir du gradient d'intensité lumineuse de l'image, a) image du gradient d'intensité et b) attracteurs (verticalement) qui vont créer le potentiel externe à partir de cette image.</u>	26
<u>Figure 2.2. Lignes perpendiculaires au contour actif pour la recherche de minima d'énergie.</u>	26
<u>Figure 2.3. Suivi d'une piste d'atterrissage depuis une caméra embarquée sur un avion.</u>	28
<u>Figure 2.4. B-Spline cubique avec un point de contrôle P de multiplicité un, deux et trois respectivement.</u>	29
<u>Figure 2.5. Zones potentielles pour la recherche de : (a) lignes droites, et (b) croisement de lignes droites.</u>	30
<u>Figure 2.6. Suivi avec modèles de contours actifs déformables d'une affiche sur un mur.</u>	32
<u>Figure 2.7. Espaces couleur et gradients: a) image d'intensité, b) image couleur (RGB), c) espace d'Ohta, d) espace de couleurs normalisées, e), f), g), et h) sont les gradients pour les images au-dessus respectivement.</u>	34

Figure 2.8. Caractérisation et détection de régions à partir des résultats de la méthode de segmentation : a) caractérisation dans l'espace de Ohta b) gradient d'intensité dans cet espace, c) caractérisation dans l'espace de couleurs normalisés, d) gradient d'intensité dans cet espace.....	35
Figure 2.9. Résultats du suivi d'une route : a) exemple d'image avec les zones segmentés utilisé pour la caractérisation de paramètres pour le suivi de contours, et de b) à h) suivi de la région par des contours actifs.....	35
Figure 2.10. Principe de la corrélation 1D. A gauche , la ligne rouge en trait pointillé donne la position prédite de la ligne recherchée ; la corrélation s'effectue le long des segments verts de direction orthogonale à cette ligne prédite ; les points bleus sont les maximaux de corrélation dans ces directions. A droite , le modèle d'une discontinuité foncé-clair de luminance.	36
Figure 2.11. Ajustement de paramètres une ligne droite avec des grosses erreurs par moindres carrés.	37
Figure 2.12. Suivi d'une piste par corrélation 1D.	38
Figure 2.13. Suivi de lignes par corrélation 1D.....	39
Figure 2.14. Suivi d'un ensemble de points en exploitant la distance de Hausdorff.....	40
Figure 2.15. Différences de régions : (a) motif de référence, (b) variation de la région (c) image des différences.....	43
Figure 2.16. Apprentissage de A fait dans la méthode de Jurie et Dhome: définition de la région cible en haut, et variations aléatoires de la position de \square_{i_1} à \square_{i_N} . A gauche , dans le repère image. A droite , dans le repère cible.....	46
Figure 2.17. Fonctions de transformation entre le repère cible et le repère image (a) transformations entre le repère image et le repère cible au μ_0 , (b) transformation entre le repère image et le repère cible au $\mu_0 + \square\mu$	47
Figure 2.18. Transformation entre les repères, pour obtenir le mouvement apparente à partir de la différence.	48
Figure 2.19. Notre méthode d'apprentissage de la matrice d'interaction dans le repère lié à la cible. A droite , dans le repère cible. A gauche , dans le repère image. En haut , la cible. Au milieu , création du repère lié à la cible. En bas , perturbations pour l'apprentissage.	49
Figure 2.20. Cible et matrice d'interaction : a) région d'intérêt, de b) à i) valeurs de la matrice A pour les huit paramètres de mouvement, où on peut apprécier que les zones homogènes dans a) génèrent des instabilités.	51
Figure 2.21. Cible et matrice d'interaction : a) région d'intérêt, de b) a i) valeurs de la matrice A pour les huit paramètres de mouvement.	51
Figure 2.22. Suivi d'une affiche dans un couloir avec différences de motifs.....	52

Figure 3.1. Evolution d'un potentiel externe : en haut, attracteurs créés par les images mises pour cette illustration dans un plan horizontal ; en bas images de gradients qui génèrent ces attracteurs.	61
Figure 3.2. Evolution d'un contour actif dans un potentiel externe nul : a) Initialisation b) et c) résultats après quelques itérations.	62
Figure 3.3. Suivi d'une affiche avec un contour actif a) convergence du contour actif sur le quadrangle et b) erreur de suivi, mais qui satisfait les conditions de forme et de gradient.	62
Figure 3.4. Seuils utilisés pour la détection d'erreur avec la corrélation 1D, a) nombre total de points corrélés, b) distance maximale des points à l'hypothèse courante sur l'état de la cible.	63
Figure 3.5. Détection d'erreur par changement brusque de la taille de la ligne entre deux images.	64
Figure 3.6. Points corrélés dans la région de la cible.	64
Figure 3.7. Suivi d'une cible plane avec des petites occultations	65
Figure 3.8. Exemples de différents environnements utilisés, pour l'évaluation de performances.	72
Figure 3.9. Densité des variations lumineuses par unité de surface : a) image de gradients d'intensité, b) image de la température spatiale, température moyenne égal a 35.	73
Figure 3.10. Différentes cibles dans différentes conditions d'entourage et quantité de gradients par zone d'intérêt.	74
Figure 3.11. Changements d'illumination pour une même cible.	76
Figure 3.12. Cible dans un environnement relativement structuré.	77
Figure 3.13. Analyse de performances pour des variations d'échelle.	79
Figure 3.14. Rotations : de a) à h) rotations autour de l'axe perpendiculaire a l'image, et de i) a l) autour d'un axe parallèle aux colonnes de l'image.	79
Figure 3.15. Translations.	80
Figure 4.1. Modèle et paramètres de la caméra.	85
Figure 4.2. Grille d'utilité d'un ensemble d'amers pour une position donnée x_c	89
Figure 4.3. Visibilité d'un amer pour le plan xy , calculé en fonction des projections dans l'image de coins de l'amer, et des paramètres intrinsèques de la caméra.	90
Figure 4.4. Visibilité d'un amer ponctuel.	91

<u>Figure 4.5. Carte de visibilité pour des amers ponctuels.</u>	92
<u>Figure 4.6. Visibilité d'un amer plan.</u>	93
<u>Figure 4.7. Carte de visibilité pour des amers plans de dimension l (indépendamment des paramètres de la caméra : focale ...).</u>	94
<u>Figure 4.8. Ajustement de paramètres de pan et tilt</u>	95
<u>Figure 4.9. Angle entre le vecteur focal et la normal au robot.</u>	96
<u>Figure 4.10. Configuration de trois amers plans dans l'espace.</u>	97
<u>Figure 4.11. Visibilité de trois amers plans depuis différentes positions d'observation, les couleurs : rouge, vert et bleu représente la visibilité de chaque amer, donc la combinaison de couleurs donne la visibilité sur les trois amers pour une position donnée. Ici encore, les paramètres de la caméra ne sont pas pris en compte.</u>	97
<u>Figure 4.12. Visibilité des amers en considérant l'angle maximal d'ouverture du champ de vue.</u>	98
<u>Figure 4.13. Angles azimut et polar entre le vecteur \mathbf{d}_i et le vecteur \mathbf{v}_i.</u>	99
<u>Figure 4.14. Algorithme pour estimer l'utilité d'un ensemble d'amers.</u>	101
<u>Figure 4.15. Optimisation du plan perceptuel par programmation dynamique ; le graphe représente les combinaisons des cibles le long de la trajectoire. Chaque noeud est l'utilité d'une combinaison de cibles, en un point donné de la trajectoire. Les arêtes joignent les nœuds correspondant à des positions consécutives sur la trajectoire. Le plan perceptuel optimal est montré en lignes foncés.</u>	102
<u>Figure 4.16. Amers visibles sur une trajectoire et paramètres optimaux de la caméra.</u>	103
<u>Figure 4.17. Transition après la détection d'erreur d'une méthode de suivi. De (a) a (f) suivi avec différences de motifs ; en (g) et (h) suivi d'un ensemble de points.</u>	106
<u>Figure 4.18. Sélection des méthodes de suivi en fonction des caractéristiques de la cible et des erreurs dans le processus de suivi.</u>	107
<u>Figure 4.19. Transition entre méthodes de suivi pour des cibles différentes.</u>	108
<u>Figure 4.20. Diagramme des fonctionnalités visuelles.</u>	109
<u>Figure 4.21. Détection des événements visuels, dans une zone liée au suivi de ligne.</u>	110
<u>Figure 5.1. Exemples de modules à l'intérieur de la couche fonctionnelle dans l'architecture LAAS.</u>	112
<u>Figure 5.2. Schéma simplifié du contrôleur.</u>	116

<u>Figure 5.3. Deux vues de dessus produites par le simulateur</u>	117
<u>Figure 5.4. Ajustement des paramètres de la caméra. En haut, position du robot et champ de vue de la caméra dans la scène. En bas, images virtuelles générées par le simulateur et données en entrée, au module de suivi des amers</u>	118
<u>Figure 5.5. Suivi et changement de cibles pendant l'exécution d'un plan perceptuel. Simultanément, ajustement des paramètres de la caméra active</u>	119
<u>Figure 5.7. Navigation à partir du suivi des primitives</u>	123
<u>Figure 5.8. Résultats de la segmentation des régions dans une image couleur</u>	125
<u>Figure 5.9. Navigation en extérieur</u>	125
<u>Figure 5.10. Suivi de l'arrière d'un véhicule par la méthode de suivi de motifs</u>	127

Introduction générale

Les travaux présentés dans cette thèse concernent le thème des fonctionnalités visuelles qu'il convient d'embarquer sur un robot mobile, afin qu'il puisse se déplacer dans son environnement. Plus précisément ils ont trait aux méthodes de suivi visuel d'objets ou d'entités extraites d'une image, pour la navigation d'un robot mobile autonome.

Le concept de **navigation** vient directement du contexte marin, comme cela est indiqué dans la définition du mot :

Navigation n. f.

1. Action de naviguer (*Voyager sur mer, sur l'eau*). Navigation maritime, fluviale, sous-marine. Navigation côtière, au long cours. 2. Art et technique de la conduite des navires (détermination de la position et tracé de la route).

A ses jours le mot navigation s'applique dans plusieurs autres contextes, par exemple la navigation d'avions ou avions, la navigation routière, et plus récemment la navigation sur le WEB.

Afin de réaliser une navigation dans n'importe quel contexte, nous avons besoin **d'amers**, mot qui de manière similaire au mot navigation, est lié à la même culture, celle des marins emmenés à voyager sur l'eau.

Amer n. m.

MAR Tout point des côtes très visible (clocher, balise, etc.), porté sur une carte, servant de repère pour la navigation.

Dans ce contexte là, les amers sont des objets qui, du fait de leur saillance par rapport à l'environnement, servent à un marin pour se repérer lorsque le navire se déplace le long des côtes. En pleine mer, les amers visuels terrestres seront remplacés par des astres.

Nos travaux ne concernent pas bien sûr ce contexte maritime, mais de manière similaire, pour qu'un robot mobile puisse naviguer dans son environnement, il doit détecter des amers, les reconnaître, se localiser vis-à-vis du monde et déterminer ou corriger sa trajectoire pour rejoindre son but. C'est sur ce thème de la navigation référencée sur des amers que nos travaux se focalisent principalement, plus précisément

sur le suivi de ces objets ou amers pendant les déplacements du robot, cela en coopération étroite avec les autres composantes du système de navigation : localisation, interprétation, génération des mouvements, planification ...

Afin de réaliser ce suivi, nous utilisons la **vision monoculaire**. Il est évident que le système complet de navigation, pourra exploiter d'autres capteurs pour avoir une bonne perception du monde dans lequel le robot doit évoluer, y compris des capteurs proprioceptifs (odométrie, gyro) ou des capteurs 3D (stéréovision ou télémétrie laser). Nous avons validé nos travaux sur les démonstrateurs du groupe de recherche Robotique et Intelligence Artificielle (RIA) du LAAS-CNRS, qui sont tous dotés de nombreux capteurs ; mais seule la vision monoculaire est exploitée dans ce mémoire.

Globalement, nos contributions portent sur les points suivants :

- Le développement et l'amélioration de deux méthodes de suivi visuel proposées dans la littérature, méthodes fondées sur les contours actifs d'une part, sur les motifs texturés d'autre part.
- L'étude de plusieurs aspects liés au contrôle de ces fonctions de suivi. Comment planifier et séquencer plusieurs tâches de suivi ? Comment reprendre des erreurs ? Pour ce faire, nous exploitons une caractérisation des méthodes intégrées dans notre système.

Enfin, l'intégration de ces méthodes dans un système complet de navigation, et particulièrement, l'analyse des modules visuels nécessaires dans un tel système.

Pour commencer dans le chapitre 1 , nous allons circonscrire nos travaux d'une manière générale, à l'intérieur de la problématique de la navigation autonome d'un robot mobile, pour ensuite aborder la problématique spécifique de la navigation dans deux environnements, qui seront traités dans cette thèse. D'abord, nous commencerons par la navigation en milieu intérieur, avec deux approches différentes qui sont: a) la navigation métrique basée sur une localisation du robot par rapport à un modèle global de l'environnement, et b) la navigation fondée sur une carte topologique. Ensuite, nous évoquerons la navigation d'un robot en environnement extérieur, plus précisément dans un environnement semi structuré : le robot se déplace le long d'une route ou d'un chemin. Ce contexte concerne deux types d'application sur lesquelles le groupe RIA réalise des études : les véhicules intelligents (suivi des autres véhicules pour l'évitement d'obstacles ou la gestion des interdistances) ou l'automatisation de machines agricoles (suivi d'un chemin pour contrôler les mouvements de la machine). Par la suite, nous ferons un rapide état de l'art sur les méthodes de suivi visuel d'objets ; il s'agira plus précisément de présenter les différentes classes de méthodes proposées dans la littérature très fournie dans ce domaine. Enfin, nous finirons ce chapitre par une description générale du système de navigation dans lequel nous voulons intégrer un module de suivi. Une contribution est l'étude globale des modules de vision intégrés dans un tel système : comment faire coopérer ces modules pour garantir une certaine robustesse ? Nous présenterons l'architecture générale du système : plusieurs instanciations seront proposées dans le chapitre 5.

Dans le chapitre 2 , nous allons décrire en détail les quatre méthodes de suivi visuels que nous avons utilisées dans nos travaux ; trois entre elles ont été développées dans le cadre de cette thèse. Nous avons tout d'abord développé une méthode fondée sur les contours actifs ou *snakes*. Elle permet l'extraction, puis le suivi de contours fermés, typiquement celui de la silhouette de l'objet d'intérêt. Nous avons proposé

l'incorporation de nouvelles forces internes, afin: a) d'adapter d'une meilleure manière, l'évolution de la courbe aux contours de l'objet à extraire de l'image, et b) d'imposer des contraintes afin que cette courbe ait une forme prédéfinie. Ensuite, nous détaillerons les études faites dans les espaces couleurs afin d'exploiter cette sorte d'information dans les fonctions de suivi par les contours actifs.

Nous avons ensuite développé la méthode de corrélation 1D, adaptée pour réaliser le suivi de lignes ou de segments de droite. Nos travaux en ce domaine ont consisté à adapter la méthode utilisée par J.B. Hayet [51], pour la localisation des amers plans ; la fonction proposée pour recalculer les arêtes projetées de l'amer sur les contours réels dans l'image, a été exploitée pour le suivi de lignes dans une séquence d'images.

Nous avons intégré dans notre système, et validé sur nos séquences, la méthode de suivi d'une région, représentée par un ensemble de points d'intérêt. Cette méthode fondée sur la distance de Hausdorff, a été développée précédemment dans le groupe par R. Murrietta [81] et V. Ayala [6] à partir des travaux présentés dans [52].

Enfin, pour finir, pour suivre une région texturée, nous avons développé la méthode de suivi de motifs planaires, proposée par Belhumeur [48] et Jurie [60] ; nous proposons un changement de repère pour l'apprentissage de la matrice d'interaction, grâce auquel la comparaison et la combinaison de plusieurs matrices pourront être faites.

Par la suite, dans le chapitre 3, nous allons caractériser chacune des méthodes décrites dans le chapitre précédent d'une manière qualitative. Nous allons détailler aussi les mesures proposées afin de caractériser les cibles et les environnements pour lesquels chaque méthode donne de bons résultats. Finalement un protocole d'évaluation a été mis en œuvre afin de déterminer la robustesse de ces méthodes lors du suivi d'une cible, selon différentes situations relatives caméra/cible (rotation, échelle), selon la nature de l'environnement dans lequel évolue cette cible (texturé, uniforme) et selon la vitesse du mouvement apparent de la cible dans l'image.

Le chapitre 4 est consacré à des aspects liés au contrôle des fonctions de suivi. Tout d'abord, nous présenterons une méthode originale de planification des fonctions de suivi qu'un robot doit exécuter lors d'un déplacement le long d'une trajectoire planifiée par ailleurs. Nous avons repris, intégré et étendu les travaux proposés dans le groupe RIA par J.B. Hayet et V. Ayala, pour la détermination des critères de visibilité d'un amer plan, et pour la génération des plans perceptuels ; notre proposition pour générer ces plans, se fonde sur une méthode d'optimisation globale ; elle permet de prendre en compte la taille et l'orientation des amers. Nous continuerons par détailler les diverses stratégies mises en œuvre pour la collaboration entre les méthodes de suivi. Ces interactions entre méthodes sont inévitables afin d'assurer la continuité du suivi dans deux contextes : (a) la transition dans le plan perceptuel, pour passer d'un amer au suivant, et (b) la reprise d'une erreur en cas de perte de la cible par la méthode chargée de la suivre. Les stratégies proposées tiennent compte de la caractérisation des méthodes ainsi que du plan perceptuelle en cours d'exécution.

Le chapitre 5 est consacré à la validation expérimentale des approches proposées dans cette thèse. Nous commencerons avec la description de l'intégration au module de suivi, de la commande d'une caméra active, dotée de modalités programmables : site, azimut et zoom. Les caméras de ce type, disponibles sur le marché, ont une interface

assez pauvre, ce qui rend difficile la mise en œuvre de lois de commande évoluées. Notre système de commande exploite la logique floue.

Enfin, nous détaillerons les expérimentations effectuées sur les démonstrateurs robotique, afin l'intégration de nos méthodes de suivi à un système complet de navigation. Nous avons d'abord conçu un environnement de simulation, qui prend en compte le temps de manière explicite ; les stratégies de commutation entre méthodes ont d'abord été mises en œuvre en simulation. Puis, nous avons réalisé plusieurs expérimentations, dans plusieurs contextes : a) le suivi d'amers plans pour la navigation fondée sur la localisation explicite du robot en milieu intérieur, b) le suivi de primitives pour la navigation visuelle fondée sur une carte topologique d'un milieu intérieur, et c) le suivi de route ou de chemin en milieu extérieur. Dans chaque cas, les interactions avec les autres composantes du système, sont exhibées et analysées.

Nous avons aussi validé nos méthodes de suivi, sur des séquences acquises depuis une voiture en déplacement le long d'une route (séquence acquise dans le cadre du projet ARCOS, Action de Recherche sur la Conduite Sécurisée).

Enfin, nous présenterons une conclusion générale et évoquerons quelques perspectives de ces travaux.

Notations :

dans la suite de cette thèse, les variables en caractères **gras** seront considérées comme des vecteurs ou matrices, et les variables en caractères *italiques* comme des valeurs scalaires.

Chapitre 1

Vision Dynamique et Navigation d'un robot mobile

Dans ce chapitre, nous introduisons le contexte de nos travaux concernant la conception et la réalisation de fonctionnalités visuelles embarquées sur un robot mobile, plus précisément les fonctions de suivi d'objets. Nous précisons tout d'abord l'utilité de ces fonctionnalités pour traiter de la navigation sur un robot mobile. Nous décrivons ensuite plus précisément les aspects spécifiques aux deux principaux contextes applicatifs : la navigation en milieu intérieur (robots de service, robots domestiques...) et la navigation en milieu extérieur (véhicules intelligents, robotique agricole...).

Enfin, nous rappelons les outils développés dans le groupe Robotique et Intelligence Artificielle du LAAS-CNRS, pour intégrer des traitements de diverses natures dans une architecture décisionnelle embarquée sur un robot mobile. Nous indiquons dans cette architecture, les fonctionnalités que nous avons utilisées et leurs interactions avec notre module de suivi. Dans le chapitre 5, nous reprendrons cette description et en donnerons plusieurs instanciations selon l'application.

1.1 Objectif de nos travaux

La robotique mobile est un champ de recherche très varié, du fait principalement des nombreuses applications potentielles. Afin de réaliser des tâches dans le cadre d'une de ces applications, un robot doit posséder la capacité de naviguer dans son environnement.

La navigation d'un robot mobile est un des problèmes clés dans la communauté de la robotique. Grossièrement, le problème consiste à intégrer sur un robot réel, toutes

les fonctions nécessaires pour qu'il puisse exécuter des déplacements à travers un environnement donné, en utilisant l'information perçue par ses capteurs. Il existe plusieurs approches pour résoudre cette tâche, mais entre toutes ces approches, les plus connues et aussi les plus utilisées sont fondées sur la construction d'un modèle du monde à partir de l'information sensorielle ; le robot utilisera ce modèle pour exécuter des mouvements requis par une tâche de plus haut niveau.

Par exemple, en milieu intérieur, le robot pourra recevoir un ordre du type « *Aller dans la pièce C* ». Le modèle de l'environnement sera exploité que cela soit afin de se localiser (« *Je suis dans la pièce A* »), pour choisir un chemin dans une description topologique (« *Pour aller dans la pièce C depuis la pièce A, je dois traverser la pièce B, puis le couloir T* »), pour générer une trajectoire (« *Pour traverser la pièce B, je dois suivre telle courbe sur le sol, définie dans un repère R_B* », « *Pour traverser le couloir T, suivre les murs ...* ») ou pour contrôler l'exécution des déplacements planifiés (« *Pour passer de la pièce B au couloir T, chercher la porte, la franchir, ...* »).

Le problème de la navigation peut donc être décomposé en plusieurs tâches, comme par exemple : la planification de trajectoires, l'évitement d'obstacles, etc., la partie de la navigation qui nous intéresse dans le cadre de cette thèse, est plus précisément la localisation automatique du robot dans le monde.

Le modèle du monde peut être représenté de différentes manières et avec différents niveaux de détail ; ces propriétés du modèle dépendent principalement de la tâche à réaliser par le robot, du type d'environnement, ainsi que de la résolution et du champ de vue des capteurs montés sur le robot. Entre tous les capteurs possibles dont un robot mobile peut disposer, comme par exemple, les capteurs ultrasons, les télémètres laser, les odomètres..., ce sont les capteurs de vision, les caméras, qui fournissent au robot le plus d'information sur l'environnement.

La vision est choisie comme capteur privilégié du fait de la richesse de l'information perçue : résolution, information aussi bien géométrique que radiométrique (luminance, couleur)... Bien que tous les robots expérimentaux disponibles dans le groupe Robotique et Intelligence Artificielle du LAAS-CNRS soient équipés de banc stéréoscopique, nous n'exploitons que la **vision monoculaire**, et tirons parti, quand cela est disponible, de la **couleur** (en milieu extérieur) et des modalités contrôlables sur la caméra (**vision active**). La richesse de l'information visuelle rend possible une interprétation contextuelle ; par exemple, dans des environnements humains, la plupart des informations exploitées par l'homme lors de ces déplacements est visuelle; une interprétation contextuelle du contenu des images, fondée sur des modèles d'entités susceptibles d'être reconnues, est indispensable à l'heure de naviguer dans cette sorte d'environnements ; citons par exemple dans le contexte de la conduite automobile, la reconnaissance des feux ou des panneaux de signalisation.

La richesse des informations contenues dans une image, mais aussi les artefacts qu'elle peut contenir (reflets, zones saturées, occultations ...), font que la tâche de reconnaissance et d'interprétation de l'information visuelle, fondée sur des critères globaux, est extrêmement compliquée, et fournit un résultat imprécis (au sens des erreurs de localisation) et incertain (au sens des possibles ambiguïtés, des erreurs de reconnaissance). Il est de ce fait, très important et nécessaire d'avoir un module de suivi d'objets, embarquée sur le robot, afin de réduire la complexité de la recherche de l'information pertinente à chaque instant pendant l'exécution d'un déplacement le long d'une trajectoire ou pendant l'exécution d'une commande référencée sur la vision. La

fonction principale d'un module de suivi, est de traiter une séquence d'images et d'exploiter sur chaque image, des critères locaux (traitement sur une région d'intérêt) afin d'identifier une entité déjà perçue et reconnue, dans les images précédentes.

Le module de suivi doit coopérer avec un module d'interprétation qui analyse toute l'image, pour initialiser au début la position des entités suivies, et pour recalculer cette position en cours de suivi ; il permet aussi de diminuer les imprécisions et incertitudes de la phase d'interprétation.

Nous résumons l'ensemble des fonctions évoquées dans cette section dans la figure 1.1. Nous nous intéressons dans cette thèse aux fonctionnalités de suivi d'objets nécessaires pour traiter principalement de la localisation visuelle d'un robot dans son environnement. Cette localisation nécessite la reconnaissance d'entités (objets, amers, primitives visuelles...), puis le suivi de ces entités lors des mouvements. Nous nous sommes essentiellement intéressés à des entités **statiques** et **rigides**.

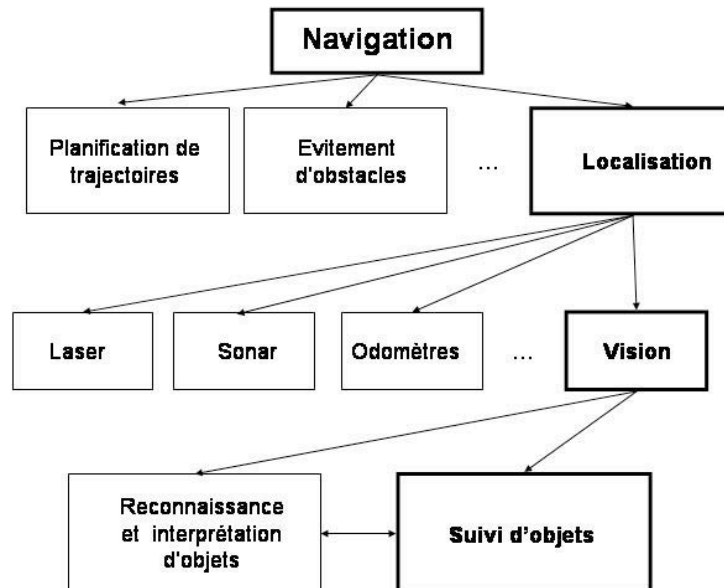


Figure 1.1. Schéma simplifié de tâches à l'intérieur du problème de la navigation autonome.

Afin d'analyser les travaux faits en vision pour la navigation d'un robot mobile, nous pouvons considérer deux grandes classes d'applications : a) la navigation en milieu intérieur et b) la navigation en milieu extérieur. Cette division classique vient principalement du fait que les problèmes pour les fonctions visuelles d'interprétation et de suivi dans chaque environnement sont très différents (voir des images typiques acquises lors de déplacements en figure 1.2). Cependant, comme on peut le constater très facilement, il existe toujours quelques recouvrements entre ces deux problématiques : par exemple, la reconnaissance ou le suivi d'une affiche sur le mur d'un couloir, ou d'un panneau de signalisation au bord d'une route, peuvent exploiter les mêmes méthodes.

Nos travaux s'appliquent aussi bien à des environnements d'intérieur (suivi des bords d'un couloir, d'une porte... suivi d'amers de type affiches...), que d'extérieur (suivi des bords d'une route, suivi d'un autre véhicule).



Figure 1.2. Différents environnements pris en compte pour la navigation a) en milieu intérieur et b) en extérieur.

Avant de présenter nos propres travaux dans les chapitres suivants, nous allons d'abord définir ces environnements intérieur et extérieur, ainsi que les conditions et problématiques implicites à chacun ; nous allons aussi décrire quelques travaux faits dans ces deux contextes. Puis, nous présenterons un bref état de l'art sur le suivi d'objets, essentiellement pour situer les méthodes que nous avons utilisées ou développées.

1.2 Navigation en milieu intérieur

La navigation en milieu intérieur concerne principalement des environnements humains hautement structurés (figure 1.2a), tel que des bâtiments publics ou des ateliers. Même s'il existe une tendance architecturale à construire des bâtiments publics plus incorporés au milieu naturel, afin d'éviter le stress et l'anxiété produits par ce style de bâtiments (par exemple en évitant les couloirs étroits et les longs murs remplis de portes), nous allons traiter dans cette thèse, des environnements intérieurs classiques, bien illustrés par nos laboratoires : un réseau de couloirs qui desservent des pièces.

Afin de naviguer dans ces environnements, le robot doit, soit construire son propre modèle du monde, soit l'acquérir par une autre méthode, comme par exemple à partir d'un modèle CAD donné par un utilisateur. En plus, il existe plusieurs possibilités pour les représentations utilisées dans ce modèle du monde, parmi lesquelles on peut citer :

- a) une représentation purement géométrique, généralement éparse (par exemple, un ensemble de segments laser),
- b) une représentation discrète de type carte d'occupation, pour décrire l'espace libre dans chaque pièce et,
- c) une représentation topologique qui décrit les connexions entre couloirs et pièces.

Les méthodes utilisées pour réaliser la navigation autonome peuvent être fondées en certains lieux (espaces ouverts) sur les deux premières ou peuvent exploiter en d'autres lieux seulement la troisième (réseau de couloirs). Nous sommes intéressés

pour intégrer des fonctions de suivi d'objet dans ces deux méthodes, que nous allons décrire plus précisément.

1.2.1 Navigation métrique.

La navigation métrique exploite une représentation géométrique du monde, qui peut être donnée par un utilisateur, ou construite par le robot lui-même, par différentes méthodes traitant du problème du SLAM *Simultaneous Localization and Mapping* [99], [80],[103]. La figure 1.3 montre une carte métrique des segments laser appris par le robot DILIGENT à partir des données laser.

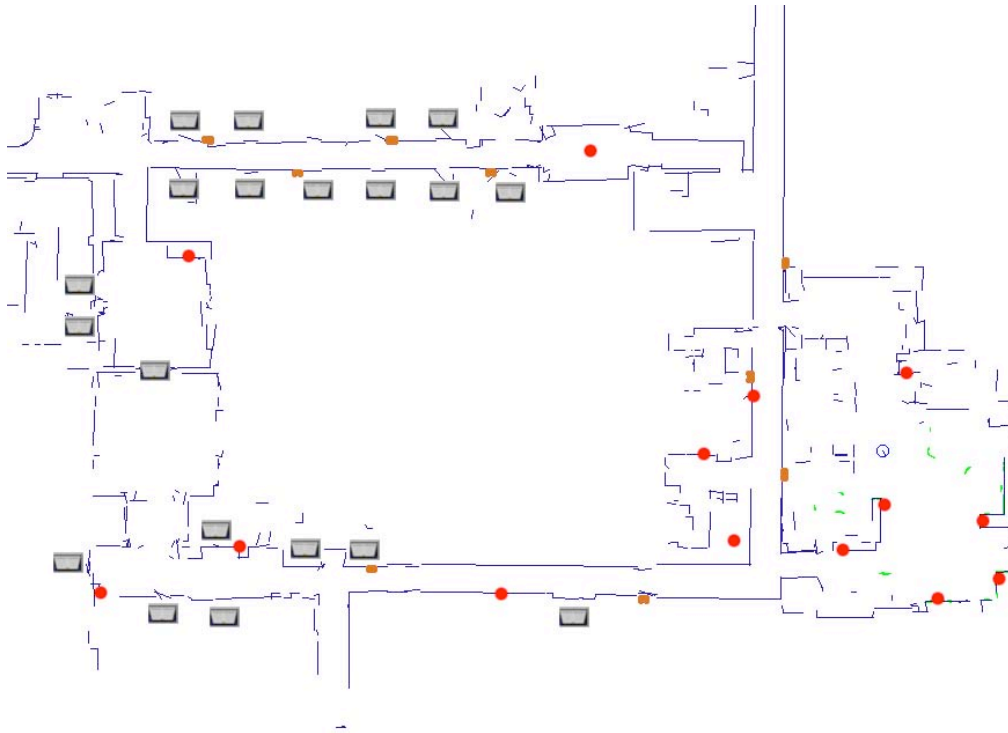


Figure 1.3. Exemple d'un modèle géométrique du monde.

Ce modèle géométrique est complété par une représentation de l'espace libre, sous la forme d'un ensemble de polygones ou d'une carte discrète (*bitmap*). Avec un tel modèle, il est relativement facile d'appliquer des algorithmes pour générer une trajectoire qui conduise le robot dans l'espace de ses configurations, de $(\mathbf{x}_1, \mathbb{I}_1)$ qui est sa situation courante, à $(\mathbf{x}_2, \mathbb{I}_2)$ qui est le but.

Cependant, l'exécution de ce déplacement est réalisée par suivi de cette trajectoire sur les données proprioceptives (odomètre, gyro) qui sont sujettes à des dérives cumulatives. Pour corriger sa trajectoire, le robot doit se localiser à chaque instant afin de rendre sa situation courante, compatible avec les données perceptuelles fournies par ses capteurs ; toute divergence entre les données acquises depuis la position réelle, et les données prédites au vu de la situation estimée par odométrie, sera utilisée pour corriger ou recalibrer cette situation. Afin d'avoir une mesure plus exacte sur la situation du robot dans le monde, il peut être doté de plusieurs capteurs, comme la vision, les sonars, les télémètres lasers.... Cependant, à mesure que le nombre des capteurs augmente, la complexité des traitements nécessaires pour extraire, comparer et

corriger les informations perçues, grandit. L'intégration dans le système embarqué sur le robot, d'une méthode de suivi permet d'éviter l'extraction en continue des informations nécessaires ; comme ce sont sur des images que l'extraction d'informations peut devenir extrêmement complexe, l'intégration des méthodes de suivi visuel est souhaitable.

En milieu intérieur, beaucoup de robots exploitent représentations éparses 2D, telles que un ensemble de segments laser (figure 1.3), faciles à extraire depuis des coupes [20], [110]. Néanmoins, pour appréhender la complexité du monde 3D, il a besoin d'amers visuels [15]. On peut définir, un amer comme une entité saillante de l'environnement, qui peut être reconnue facilement. Ce sont ces amers visuels qui seront la cible du module de suivi dans la séquence d'images acquises lors des déplacements du robot.

La sélection des amers pour la navigation est un sujet très important [9] [108]; le choix des amers dépend de la tâche à réaliser ainsi que des algorithmes disponibles. Parmi les amers utilisés couramment pour la localisation d'un robot, citons les arêtes verticales [83], les zones texturées [32], ainsi que les affiches ou les panneaux de signalisation. L'utilisation de hardware spécialisé a été aussi une solution, par exemple, Hashima et al. dans [50], montrent un robot de service, chargé de porter des plateaux repas aux malades dans un hôpital : il est capable de naviguer et de se localiser dans un environnement d'intérieur à partir d'amers, définis comme les petites régions comportant des différences importantes de niveau de gris, régions qui maximisent un critère d'auto-corrélation ; de telles régions correspondent à des fenêtres entourant des points d'intérêt extraits par le détecteur de Harris. Ces amers sont détectés, puis suivis par une mesure de corrélation, avec un système de vision basé sur un processeur de corrélation de haute vitesse.

Plusieurs travaux ont été développés pour la localisation du robot à partir des amers. Dans [29], l'auteur utilise un filtre de Kalman, pour actualiser la position du robot, en exploitant la distance à un seul amer. Cependant, avec un seul amer de ce type, la position du robot dans le monde ne peut être complètement déterminée ; elle peut seulement être actualisée. Dans [72] et [74], la position du robot est trouvée à partir de la triangulation des amers visibles dont les positions ont été préalablement apprises dans le monde. Une comparaison expérimentale des méthodes de localisation peut être trouvée dans [68].

Dans des travaux récents du groupe RIA, notamment les travaux de JB.Hayet [51], des amers plans quadrangulaires et texturés, en pratique des affiches ou posters, sont automatiquement détectés, puis caractérisés afin d'être reconnus et exploités pour la localisation du robot ; de plus, le robot peut se localiser relativement à ces amers. Sur la figure 1.2a, plusieurs amers de ce type apparaissent sur les murs du couloir que le robot va devoir parcourir. Une extension pourrait être de considérer aussi des panneaux de signalisation, et d'interpréter l'information qu'ils donnent. En liaison avec JB.Hayet, nous nous sommes particulièrement intéressés dans cette thèse, au suivi visuel de cette sorte d'amers : des résultats seront présentés au chapitre 5.

1.2.2 Navigation avec une carte topologique.

Un des problèmes avec l'approche métrique est la grande quantité de données nécessaires pour mémoriser le modèle du monde, et la complexité de la procédure

permettant de comparer et corriger les informations acquises depuis tous les capteurs montés sur le robot.

Une manière plus qualitative de représenter le monde dans lequel notre robot va évoluer, semble être une bonne alternative. La représentation qualitative la plus communément utilisée, est appelée **carte topologique** ; c'est un graphe, dans lequel chaque nœud correspond à un endroit caractéristique (carrefour entre couloirs, entrées dans les espaces ouverts ...), appelé *meeting point* dans plusieurs travaux [26], que nous appellerons **lieu** par la suite. Chaque lieu devra être décrit par un ensemble des caractéristiques propres, qui permettront au robot de le reconnaître. Une arête liant deux nœuds signifie qu'il existe une commande référencée capteur que le robot peut exécuter afin de se déplacer entre les deux lieux correspondants. Cette représentation est particulièrement adaptée pour décrire des réseaux de couloirs : les lieux correspondent aux carrefours, les arêtes aux couloirs qui les lient.

Pour la génération d'une trajectoire entre deux lieux définis dans une carte topologique, il suffit de rechercher un chemin dans le graphe. Les consignes pour naviguer seront des commandes référencées capteur, par exemple *Suivre le couloir*, *Tourner à gauche*, *Franchir la porte ...* Pour chaque consigne, il existera des caractéristiques précises de l'environnement à suivre (plinthes, arête d'un mur, bords de la porte...) ; un module de suivi de primitives visuelles, sera donc requis.

Dans le cas de la navigation topologique, le robot n'a nul besoin de connaître sa position précise par rapport à un repère du monde. Il doit seulement déterminer une localisation qualitative « *Je suis dans le lieu A* » et, au mieux, une estimation de sa position relative par rapport à un repère lié au lieu dans lequel il se trouve.

De ce fait, une localisation explicite étant superflue, les lieux du monde pourront être uniquement représentés par exemple, par leur apparence. C'est alors avec la vision que l'information relevante de chaque lieu caractéristique d'un environnement intérieur peut être extraite le plus facilement. Les caractéristiques propres de chaque lieu peuvent être des simples lignes droites comme des arêtes verticales ou horizontales [34], des zones texturées [106] ou des objets particuliers (par exemple, panneaux de signalisation) [77] ; les autres capteurs (ultrasons, télémétrie laser) sont plus utilisés pour la construction du modèle métrique du monde. C'est dire que la navigation topologique est hautement dépendante d'un bon système de vision afin d'identifier les éléments caractéristiques de l'environnement.

Essentiellement, c'est à partir de correspondance entre images que les lieux sont identifiés, par des **méthodes d'indexation**. Par exemple, dans [54], les images sont comparées dans le domaine fréquentiel ; dans [17] les histogrammes couleur des images sont utilisés, et dans [84] les images sont représentées par des vecteurs d'attributs, analysées par Analyse en Composantes Principales (ACP). Plusieurs approches qui utilisent les images directement pour calculer la position du robot, utilisent des caméras omnidirectionnelles [45]. Des travaux pour la localisation qualitative en milieu extérieur, pour un robot équipé d'une caméra omnidirectionnelle ont été effectués dans notre groupe, par J. Gonzalez Barbosa [47].

Afin de gérer plusieurs hypothèses sur la possible position du robot, une approche classique, la **localisation markovienne** [98], consiste en l'accumulation d'évidence afin d'avoir une probabilité d'être en chaque lieu. Par exemple, dans [91], un

système de vote a été proposé à partir de caractéristiques trouvées dans la scène. Cette approche permet aussi d'intégrer aisément les caractéristiques trouvées par d'autres capteurs que la vision, et donc il est possible de comparer et/ou corriger la position du robot en exploitant toutes les caractéristiques acquises depuis un système multisensoriel.

1.3 Navigation en extérieur

La différence principale entre la navigation en extérieur et celle en intérieur, est que les conditions d'illumination sont non contrôlables ; quelques fois elles ne sont pas même prédictibles. Généralement les environnements d'extérieur sont plus riches en informations (couleur, textures), mais aussi plus complexes. La simplification qui consiste à exprimer le modèle du monde et la situation du robot en 2D, n'est plus possible ; le modèle doit être 3D, même si le plan du sol qui porte le véhicule (dans le cas de voitures, ou de robots mobiles à 4 roues) aura un rôle spécifique.

Pour faire face à cette plus grande complexité de l'environnement, il faut adopter des capteurs qui en donnent une description la plus riche possible : la vision s'impose généralement dans ce contexte, que cela soit en monoculaire [28], [82] ou en stéréovision [71]. Dans les travaux sur les véhicules intelligents, elle est généralement associée à un capteur télémétrique (radar, laser) pour la détection des obstacles lointains : se pose alors le problème de la **fusion multisensorielle**.

Il existe plusieurs travaux pour la navigation en environnement d'extérieur, travaux assez différents les uns des autres du fait de la variabilité de ces environnements. Cela va de la navigation de voitures sur route ou en milieu urbain, jusqu'à la navigation d'un robot d'exploration planétaire [23].

Dans le premier cas, il s'agit essentiellement de garantir que le véhicule reste sur une route, ou sur une voie de circulation, qu'il reste à distance réglementaire du véhicule précédent et qu'il détecte tout autre objet sur la route. Les scènes à traiter sont dynamiques (trafic routier) et l'environnement semi structuré (marquages sur la chaussée). Dans ce contexte, le suivi visuel concerne

- soit le bord de la chaussée, pour un contrôle automatique du véhicule.
- soit les autres objets présents sur la route (autre véhicule, piétons...), pour évaluer trajectoires (ou pistes) de ces objets et estimer leur dynamique.

Dans le cas de la navigation d'un environnement totalement naturel, par exemple pour l'exploration planétaire, le système embarqué doit intégrer un grand nombre de modules, depuis la détermination de la nature du terrain pour choisir un mode de navigation ou de locomotion, jusqu'à la détection d'amers exploités pour la localisation ou pour la commande référencée vision du robot. Dans ce contexte, le suivi visuel sera nécessaire pour accélérer la recherche des amers dans une séquence d'images.

On peut mentionner d'autres travaux pour différents contextes comme par exemple les travaux de Fernandez et Casals [42] pour la navigation en environnements non structurés, ou la navigation en extérieur sur des chemins avec interprétation de symboles en braille [115].

Dans le groupe RIA, plusieurs travaux ont été développés pour la navigation en extérieur, notamment, dans le cadre du projet EDEN pour l'exploration planétaire. Lors

de ces travaux, le suivi visuel d'amers n'a pas été considéré : les amers sont détectés sur chaque image selon des critères globaux, puis appariés en considérant une prédiction sur le déplacement effectué par le robot entre les deux points de vue. Citons des travaux en cours en robotique aérienne ; le suivi d'un amer détecté au sol depuis un robot aérien, sera plus simple à traiter du fait de l'absence d'effet perspective dans les images. Un modèle affine de la caméra peut dans ce contexte, être suffisant.

Le groupe est impliqué également dans deux actions de recherche, concernant la robotique en milieu extérieur :

- la détection d'obstacles depuis un véhicule en milieu routier (projet ARCOS).
- l'automatisation de machines agricoles, dans le cadre d'un projet financé par le Laboratoire Franco-mexicain d'Informatique (LAFMI).

De ce fait, nous présenterons aussi des résultats en chapitre 5 , sur l'application de nos méthodes de suivi sur des séquences d'images acquises depuis un véhicule sur une route, en coopération avec un collègue chargé de la détection de ces obstacles.

Pour la navigation d'une machine agricole, le problème principal consiste à décrire un environnement faiblement structuré en considérant des entités de type route ou chemin, champ, arbre puis, de planifier et exécuter grâce à un tel modèle, des trajectoires pour aller d'un lieu *A* à un lieu *B*. Le problème principal consiste à circuler dans un réseau de chemins qui permettent d'accéder à des champs ou autres espaces ouverts ; la navigation topologique et la localisation markovienne sont adaptées pour traiter ce problème. Nous présenterons aussi en chapitre 5 , l'application de nos méthodes de suivi, pour suivre les bords d'un chemin pendant les déplacements, donc en coopération avec un algorithme de détection de ce chemin [5]. La figure 1.2 (b) montre une image typique d'un tel environnement.

Nous avons appliqué nos méthodes de suivi également sur des séquences d'images acquises depuis un véhicule sur une route. Pour la navigation d'un robot en environnements naturels, mais plus précisément pour la navigation des machines agricoles, dont le problème principal consiste en trouver le chemin pour aller du point *A* au point *B*.

1.4 Suivi d'objets : Etat de l'art

Dans la littérature, de nombreuses méthodes de suivi d'objets ont été présentées ; une grande partie d'entre elles, peuvent être utilisées pour suivre des objets précis en temps réel. Malheureusement, la plupart de ces méthodes sont fondées sur un seul modèle de cible ou une seule modalité ; elles sont limitées à certains environnements spécifiques, typiquement pour la vidéo surveillance, des scènes statiques, contrôlés et connues à priori, sur lesquelles la méthode suit tous les objets dynamiques qui apparaissent.

On peut admettre, sans aucun risque de se tromper, que jusqu'à aujourd'hui il n'existe pas une méthode suffisamment générale pour traiter avec succès et robustesse, la grande variété des cibles et conditions qui existent dans le monde réel.

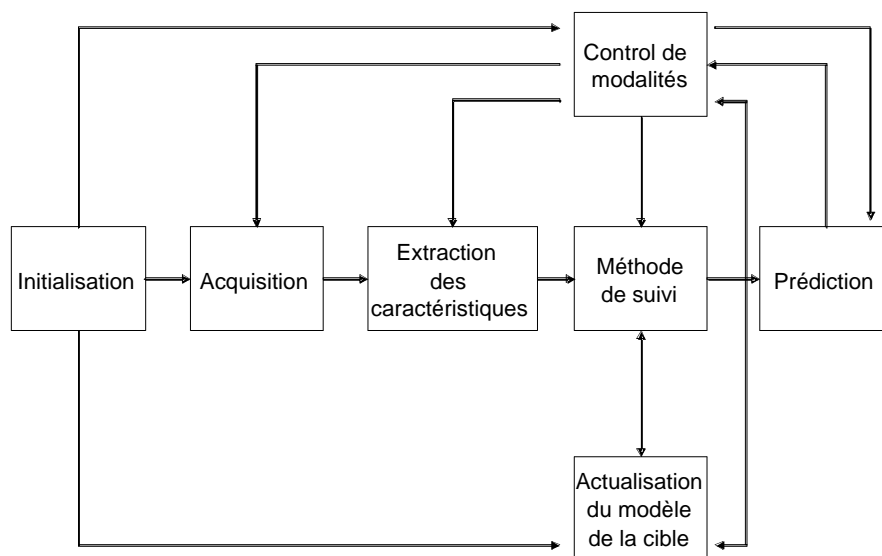


Figure 1.4. Schéma général sur le suivi.

Dans la figure 1.4, nous montrons un schéma général pour le suivi d'objets. Selon la méthode, les procédures inscrites dans ce schéma sont plus ou moins complexes : par exemple, l'actualisation du modèle de la cible ou la prédiction de son mouvement apparent dans la séquence d'images ne sont pas toujours effectuées.

1.4.1 Méthodes de suivi

Plusieurs classifications des méthodes de suivi visuel d'objets ont été proposées dans la littérature ; elles dépendent autant des auteurs, que du but pour lequel ces méthodes ont été conçues. Nous considérons la classification donnée dans [73], où selon les auteurs, les méthodes de suivi visuel peuvent être divisées en quatre classes :

- Méthodes de suivi fondées sur des modèles. Ces méthodes repèrent des caractéristiques connues dans la scène et les utilisent pour mettre à jour la position de l'objet. Parmi ces méthodes, citons celles qui exploitent les modèles géométriques fixes [18], et les modèles déformables [19].
- Méthodes de suivi de régions ou *blobs*. Cette sorte de méthodes se caractérise par la définition des objets d'intérêt comme ceux qui sont extraits de la scène en utilisant des méthodes de segmentation. Citons les nombreuses méthodes qui détectent une cible à partir de son mouvement sur un fond statique ou quasiment statique [67].
- Méthodes de suivi à partir de mesures de vitesse. Ces méthodes peuvent suivre les objets en exploitant les mesures de leur vitesse dans l'image, avec des mesures telles que le flux optique ou des équivalents [100].

- Méthodes de suivi de caractéristiques. Ces méthodes suivent certaines caractéristiques de l'objet, comme des points, des lignes, des contours ...[12], caractéristiques ou primitives image auxquelles il est possible aussi d'imposer de restrictions globales [39]. Ces caractéristiques peuvent être aussi définies par la texture ou la couleur [35].

Cette classification n'est pas exhaustive, et à ce jour, il existe de nombreux recouvrements entre les classes, c'est-à-dire, des méthodes qui peuvent être classifiées dans deux ou plus de classes. Nous considérerons que ces méthodes sont des combinaisons des approches existantes.

C'est à partir des définitions des environnements et des cibles pour chaque problématique de la navigation évoquée précédemment, qu'il est possible de s'apercevoir, que certaines méthodes, comme le suivi de *blobs*, seront difficilement utilisables. Par ailleurs, il est très difficile d'utiliser des méthodes fondées sur la différence objet/fond, parce que le robot est en mouvement et donc, le fond ou l'arrière-plan n'est pas statique (du point de vue de l'image), et même les cibles peuvent être statiques par rapport au fond.

De manière similaire, les méthodes fondées sur des mesures de vitesse, seront difficilement exploitables pour la navigation de robots mobiles. Il existe quelques approches pour le suivi à partir de flux optique, qui ont été essayées pour la navigation d'un robot [31]; mais, la plupart d'entre elles utilisent une méthode de suivi de caractéristiques comme des lignes droites, et c'est à partir de ces primitives éparses dans l'image, que le calcul de la vitesse est fait.

Nous favorisons donc, l'utilisation des deux autres sortes de méthodes afin de réaliser les tâches de suivi depuis un robot se déplaçant dans les environnements décrits précédemment : suivi fondé sur un modèle de la cible et suivi de primitives image. Une analyse des méthodes de suivi ainsi que des méthodes de segmentation et reconnaissance des cibles pour la navigation robotique est décrit dans [37]. Signalons qu'il existe d'autres travaux sur le suivi dans le groupe Robotique et Intelligence Artificielle,

- travaux dédiés au suivi de personnes, donc d'objets déformables (visages, mains, corps entier) [78].
- et travaux sur le suivi de points d'intérêt, algorithme inspiré d'une méthode d'appariements conçue initialement pour le recalage de vues aériennes [59].

1.4.2 Contrôle d'exécution du suivi visuel

Au delà du choix d'une méthode de suivi pour une application donnée, un autre problème consiste à contrôler son exécution : un module de contrôle du suivi devra traiter les problèmes suivants :

- Détecter l'apparition d'une nouvelle cible, et initier un nouveau suivi, éventuellement en commutant d'une cible à une autre (typiquement, le long d'un couloir).

- Détecter une erreur et initier une procédure de reprise, ce qui peut nécessiter de coopérer avec un autre module visuel : soit un module moins réactif, chargé de retrouver la cible dans l'image selon des critères globaux, soit une autre méthode de suivi, plus adaptée ou plus robuste.
- Commander des modalités contrôlables sur la caméra, afin d'adapter le champ de vue aux mouvements de la cible : en ce cas, le module de suivi est couplé avec un module d'asservissement, typiquement pour commander rotation en site et azimut ou zoom sur une caméra active de vidéo surveillance.

Comme nous l'avons mentionné, dans des environnements réels, une seule méthode de suivi ne peut pas traiter toutes les tâches et tous les contextes qui peuvent se présenter. D'ailleurs les méthodes de suivi ne sont généralement pas suffisamment robustes pour travailler avec des conditions variables de l'environnement ; elles échouent fréquemment quand les conditions d'illumination changent significativement.

D'autres situations caractéristiques qui provoquent des erreurs pour les méthodes de suivi, concernent des arrières plans ou fonds encombrées, des changements de pose de la cible, des occultations de la cible ou des fortes discontinuités dans sa dynamique. Certains de ces problèmes peuvent être surmontés en utilisant un module de prédiction adapté, capable de maintenir la position de la cible pendant les occultations, ou capable de gérer plusieurs modèles pour la dynamique de la cible (vitesse constante, accélération constante et autres) ou pour sa trajectoire (ligne droite, courbes ...), ou capable de gérer plusieurs hypothèses sur la position de la cible dans l'image (filtrage particulière). Dans nos travaux, par manque de temps, nous n'avons pas exploité ces techniques, mais il est évident que leurs contributions permettent de rendre plus robustes les méthodes de suivi évoquées ici.

Nous avons exploré une autre voie afin de surmonter les limitations des méthodes de suivi : l'intégration de plusieurs méthodes afin de les faire collaborer entre elles. Dans ce cas, un module de contrôle d'un module de suivi intégrant plusieurs méthodes, doit être chargé de sélectionner la méthode à utiliser en fonction de la cible et des conditions de l'environnement. En plus, quand une méthode spécifique a échoué, le module peut sélectionner une autre en fonction du problème, afin de récupérer la cible, et si ce n'est pas possible, afin de sélectionner une nouvelle cible, laquelle dépendra de la tâche spécifique à exécuter.

L'intégration de plusieurs méthodes de suivi est destinée également à augmenter la robustesse du système de vision robotique. Toyama et Hager [105], ont proposé d'utiliser une hiérarchie en couches pour les méthodes de suivi. D'autres auteurs ont proposé d'utiliser une fusion de l'information de chaque méthode de suivi pour améliorer la robustesse de l'ensemble des méthodes. Kruppa et al. [69] utilisent une approche de d'information réciproque, afin de fusionner différents modèles de la cible, obtenus depuis plusieurs méthodes de suivi exécutées simultanément. Rasmussen et Hager [90] utilisent une approche probabiliste pour contrôler une hiérarchie de stratégies de méthodes de suivi. Jepson [58] propose d'intégrer plusieurs méthodes en limitant leur champ d'application pour le suivi de visages dans des environnements naturels encombrés. Adachi [1] renforce la performance des méthodes de suivi en améliorant le modèle de mouvement affine.

D'autres travaux ont été focalisés sur le contrôle de plusieurs méthodes de suivi d'une manière active. Barretto [11] a développé des mesures pour l'évaluation des performances afin de contrôler le suivi actif en utilisant une tête stéréo.

Dans les chapitres 2 et 3, nous décrivons et analyserons les méthodes de suivi que nous exploitons dans nos travaux ; dans le chapitre 4, nous verrons comment nous les faisons coopérer afin de rendre le suivi plus robuste et afin d'accroître les capacités de navigation de nos robots en considérant différents types d'amers visuels.

Un point important à prendre en compte, dans les besoins d'un système de suivi d'objets est l'intégration avec un module de *vision active*, chargée de contrôler des modalités sur le capteur lui-même. Ce contrôle actif du capteur permet d'améliorer le temps de réponse du suivi, par focalisation de l'attention sur des zones d'intérêt. Nous allons donc décrire les caractéristiques de la vision active et dans le chapitre 5 nous décrivons nos travaux sur la commande d'une camera active.

1.5 Vision Active

Plus d'un tiers du cerveau humain est dédié au processus de la vision [43]. Pas surprenant, puisque ce processus est la méthode la plus importante par laquelle le cerveau acquiert des informations sur le monde extérieur ! La vision chez l'homme est un processus dynamique pendant lequel les yeux échantillonnent en continu leur environnement. C'est une des raisons qui font qu'il est très important d'avoir un tel système de **vision active** chez les robots.

C'est à partir du développement d'ordinateurs de plus en plus rapides et avec de plus grandes capacités de stockage, qu'il est devenu possible d'embarquer des fonctions de vision active sur des machines. Contrairement à l'idée initiale de vision passive [76], utilisé au début de la robotique et principalement dans les premières approches de la vision par ordinateur, la vision active [18] s'intéresse au contrôle des algorithmes ainsi que des structures optiques et mécaniques des caméras, afin de simplifier (a) le processus de la vision par ordinateur et en même temps, (b) l'interaction avec le monde extérieur. Il est important de remarquer qu'un système de *vision active*, crée lui-même des processus actifs dans le monde.

Citons les caractéristiques essentielles d'un tel système :

- *opération en continu*. Le système est toujours en marche.
- *filtre d'information*. Un système de vision active agit comme un filtre d'informations, ne retenant que ce qui est pertinent pour la fonction visuelle en cours d'exécution ; c'est la notion de **purposive vision** introduite par Y.Aloimonos [3] [8].
- *temps réel*. Afin d'être utile, un système de vision active doit retourner les résultats dans un délai fixe, qui dépend de l'application.
- *contrôle du traitement*. Afin de réaliser des fonctions en temps fixe ou au moins, borné, le traitement est restreint à des régions d'intérêt. ; le capteur est configuré pour faciliter le choix et la taille de cette région (centrage, zooming.... pour garantir une résolution fixe).

Pour qu'un robot puisse agir et intervenir dans un monde dynamique, par nature très complexe, il est vital qu'il ait la capacité de commander activement les paramètres de la caméra. Les systèmes de vision active doivent donc avoir des mécanismes pour commander les paramètres tel que l'orientation, le zoom, l'ouverture et la convergence (pour la stéréovision) en réponse aux besoins de la tâche et aux stimulus externes.

Plus généralement, la vision active renferme des mécanismes d'attention, de perception sélective dans l'espace, la résolution et le temps. Ces mécanismes sont introduits en modifiant soit les paramètres physiques de la caméra, soit la manière avec laquelle les données sont traitées après acquisition.

L'étroit accouplement entre perception et action, proposé dans le paradigme de la vision active, ne se limite pas au contrôle du processus visuel. Ce processus doit aussi interagir fortement avec les activités qui exploitent les données visuelles (la navigation, la manipulation, la signalisation du danger, etc.) ; ceci permet des simplifications dans les algorithmes de commande et les représentations de scènes, des temps de réaction rapides, et finalement, des taux de succès plus élevé pour ces activités.

L'application de la vision active facilite certaines tâches qui seraient impossibles en utilisant la vision passive. L'amélioration de la performance peut être mesurée en termes de fiabilité ou robustesse, de répétabilité et de réactivité (ou temps d'exécution) lorsque sont traitées certaines tâches spécifiques, mais aussi par la généralité, ou capacité à traiter une grande diversité de tâches.

1.6 Architecture du système - l'architecture LAAS

Dans la figure 1.5, nous montrons l'architecture LAAS, architecture des systèmes embarqués sur tous les démonstrateurs du groupe RIA. Nous avons intégré différents modules dans cette architecture, pour les démonstrateurs DILIGENT et DALA (figure 1.6).

Le système embarqué sur les robots, compte trois niveaux hiérarchiques distincts. Nous ne sommes intervenu que sur le niveau fonctionnel :

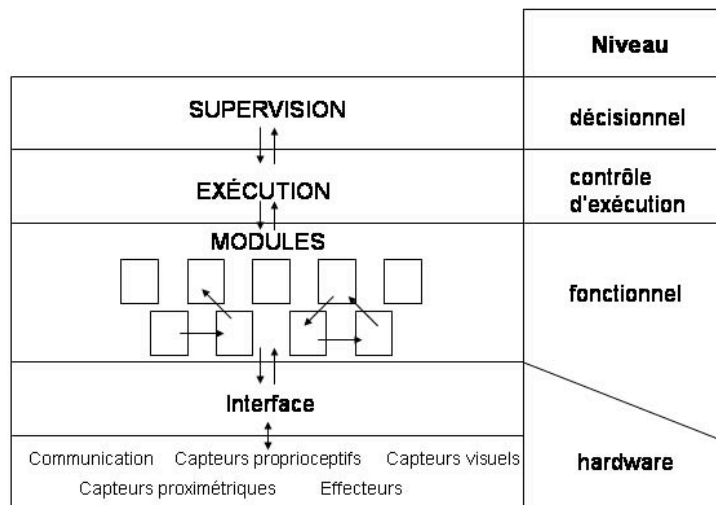


Figure 1.5. L'architecture LAAS: vue d'ensemble

- *Niveau Fonctionnel.* Ce niveau inclut les traitements nécessaires pour toutes les capacités sensori-motrices de base du robot, ce qui comprend aussi bien le traitement des données sensorielles, que la commande des actionneurs. Ce niveau contient une bibliothèque de *modules*, lesquels sont des entités autonomes, de natures très diverses, réalisant chacun une série de fonctionnalités spécifiques et dotées entre elles de capacités de communication, soit par messages (interaction de type *requête/réplique* entre un module client et un module serveur), soit par mémoire commune (un module serveur produit une donnée dans un *poster* lu par des modules clients).
- *Contrôle d'exécution.* Ce niveau permet de contrôler et coordonner l'exécution des fonctions du niveau précédent, suivant les besoins et contraintes exprimés par le niveau suivant ;
- *Niveau Décisionnel.* Ce niveau contient les capacités du robot à planifier les tâches requises par une mission donnée par un opérateur. Il supervise l'exécution de ces plans, de telle sorte à pouvoir réagir aux événements imprévus par les planificateurs.

Dans [2], on peut trouver une description beaucoup plus détaillée de l'architecture LAAS et des outils développés pour la mettre en œuvre.



Figure 1.6. Robots démonstrateurs du groupe RIA : a) Diligent, b) Dala.

Comme il a été mentionné, nous nous intéressons aux fonctionnalités visuelles de suivi et à leur contrôle, ce qui va impliquer des coopérations ou interactions entre plusieurs fonctionnalités : par exemple, entre deux modules de suivi, entre un module de segmentation et un module de suivi, entre un module de détection d'amer et un module de suivi. Les algorithmes et/ou fonctionnalités diverses se trouvent encapsulées dans des modules autonomes dans le niveau fonctionnel. Un mécanisme de *requêtes*, permet à un autre module, au superviseur, voire à l'utilisateur en personne de démarrer, d'interrompre une fonction, ainsi que de changer des paramètres internes au module. Ces requêtes peuvent s'exécuter en parallèle, de manière synchrone ou non. Les modules génèrent des *répliques* et des *bilans*, qui permettent de savoir comment s'est déroulée l'exécution des requêtes et le cas échéant, et de faire remonter la cause des erreurs observées. L'exportation de données internes d'un module se fait par les structures publiques appelées *posters* : c'est via ces structures qu'un module de segmentation pourra transmettre le modèle de la cible (contour, région) au module de suivi, ou réciproquement, que le module de suivi pourra fournir une prédiction sur la position d'un amer à un module de localisation du robot.

Nous présentons en figure 1.7, l'architecture générique de notre système de suivi visuel. Il y apparaît trois modules principaux :

- Module de contrôle du suivi : Module chargé de la sélection des cibles ou objets que le robot doit suivre. Les cibles peuvent être inconnues et détectés en ligne, ou déjà connues et leurs modèles enregistrés; leur suivi peut être prévue déjà un plan perceptuel.
- Module de suivi : Module dans lequel se trouvent toutes les fonctions utiles pour le suivi visuel. Cela peut inclure plusieurs méthodes de suivi (régions ou contours), et aussi, des fonctions d'auto-détection d'erreurs.
- Module d'extraction: Module qui interagit avec le module de suivi afin de donner l'initialisation de l'objet ou cible à suivre, qui sera extrait d'une image peut être par segmentation ou reconnaissance d'objets 3D, localisation...

Les contraintes temporelles sont fortes surtout pour le module de suivi, qui typiquement devra fonctionner à une cadence de 10 à 50HZ. Nous supposons que le module d'extraction peut avoir un temps de réponse 10 fois plus important.

Le module de contrôle du suivi, est chargé de répondre aux sollicitations des couches supérieures du système. Il coordonne l'exécution d'un plan perceptuel, donc, il lance des requêtes de démarrage ou arrêt d'activité. Par exemple une fois la caractéristique désirée trouvée par le module d'extraction, il va demander au module de suivi de suivre telle cible, puis recevra de l'un et de l'autre des bilans issus de la comparaison de leurs résultats ; s'il détecte une divergence, il est responsable de la stratégie pour reprendre l'erreur.

La communication entre le module de contrôle et les autres se fait principalement par requêtes et répliques et/ou bilans ; entre le modules de suivi et celui qui détecte les cibles, elle se fait principalement par moyen de l'exportation de posters, contenant l'information à partager. Ces modules peuvent interagir avec d'autres modules (localisation du robot, asservissement visuel...), mais c'est principalement ce schéma qui nous intéresse. Dans le Chapitre 5 , nous donnerons plusieurs instanciations des modules qu'il contient..

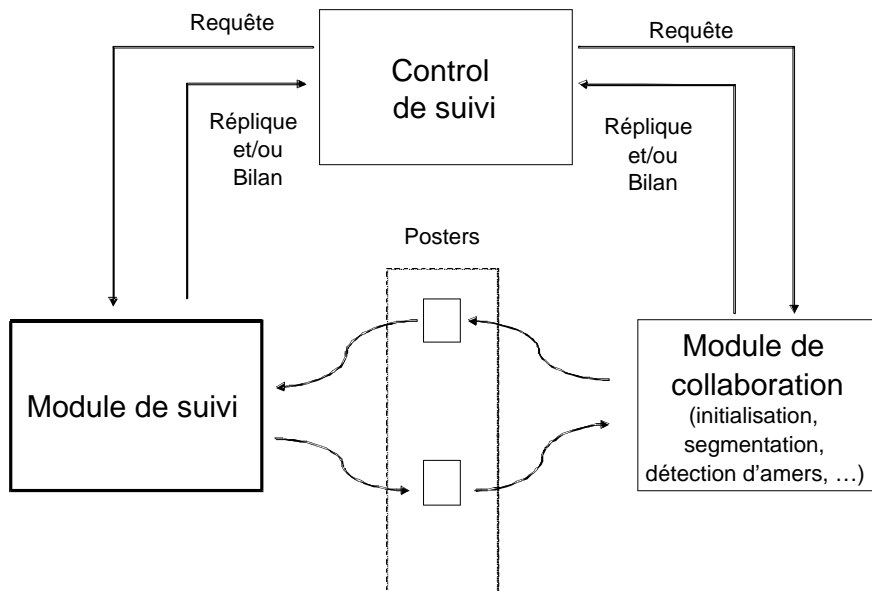


Figure 1.7. Architecture générique du système visuel.

1.7 Conclusions

Dans ce chapitre nous avons présenté d'une manière générale le contexte de nos travaux, qui concernent les fonctionnalités visuelles pour la navigation de robots autonomes dans des environnements autant d'intérieur et que d'extérieur. Ces fonctionnalités visuelles seront plus spécifiquement le suivi d'objets dans des séquences d'images acquises depuis une machine ou un robot en mouvement, et son application à la navigation d'un robot mobile dans les types d'environnement.

Un panorama général de l'architecture des systèmes embarqués sur les démonstrateurs robotiques du LAAS-CNRS, a été montré ainsi que la place dévolue au

module de suivi visuel, et les principales collaborations qu'il devra avoir avec les autres modules du système embarqué.

Un bref état de l'art a été présenté, sur les méthodes de suivi visuel, ainsi que les différentes approches proposées pour la collaboration de différentes méthodes afin d'améliorer la performance du module de suivi.

Chapitre 2

Méthodes de suivi visuel

2.1 Introduction

Le suivi visuel d'objets dans des séquences d'images est devenu une des tâches les plus utilisées, parmi celles développées pour la vision par ordinateur. Les différents champs d'application comprennent entre autres : la vidéoconférence (suivi des interlocuteurs), la vidéosurveillance (détection, suivi, reconnaissance du comportement de personnes, d'intrus), la segmentation et l'édition de séquences vidéo, sans oublier les applications plus traditionnelles de la robotique : suivi d'obstacles pendant une phase d'évitement, suivi d'indices visuels dans une tâche asservie sur la vision, suivi d'un opérateur (corps, visage, main...) pour définir des modes d'interaction évolués entre l'Homme et la Machine...

Le but central du suivi visuel consiste à déterminer la configuration d'une cible ou les caractéristiques d'un objet dans une séquence d'images, tant que cette cible est en mouvement apparent dans les images ; ce mouvement dans les images peut provenir d'un réel mouvement 3D de la cible dans le champ de vue de la caméra (scénario de type vidéosurveillance), du mouvement de la caméra si celle-ci est montée sur un actionneur : robot, platine site / azimut... (scénario du robot qui suit des amers fixes durant ses déplacements) ou de la combinaison de ces deux mouvements (scénario du véhicule qui suit une cible mobile, ou qui évite un obstacle dynamique).

Comme nous l'avons dit dans le chapitre précédent, il existe différentes méthodes de suivi visuel, ainsi que différentes typologies proposées pour les classer. Si nous nous limitons aux méthodes de suivi de caractéristiques image, les principales variantes proviennent de la représentation de la cible dans l'image, et de la méthode de

prédiction sur le mouvement apparent de la cible De manière très grossière, on peut caractériser les méthodes qui suivent une cible donnée par (a) un ensemble de points, (b) une région ou (c) un contour, et qui ont une fonction de prédiction fondée sur (1) un simple filtre alpha/beta, [11] (2) le filtrage de Kalman [10] ou (3) le filtrage particulaire (ou *CONDENSATION*) [53].

Cette classification sera utilisée plus précisément dans la suite pour décrire chaque méthode et caractériser les conditions dans lesquelles elles donnent des bons résultats. Dans nos travaux, nous ne nous sommes pas intéressés à la fonction de prédiction ; un simple report du mouvement apparent trouvé à l'image précédente, nous permet de prédire la position de la cible dans l'image courante.

En robotique mobile, avoir au moins, une méthode de suivi visuel intégré sur le robot, est devenu indispensable. En plus, si le robot doit être capable de réaliser des tâches dans un environnement complexe et dynamique, il faut intégrer plusieurs méthodes dans le système embarqué. En effet, les cibles qu'un robot peut suivre dans un environnement donné, sont de natures si différentes (objet 3D ou plan, primitives, silhouettes...) et les conditions et contextes dans lesquels le suivi doit fonctionner, sont si variables, qu'une seule méthode ne peut suffire.

Dans les sections de ce chapitre, nous décrivons les quatre méthodes que nous avons intégrées sur les robots démonstrateurs exploités au sein du groupe *Robotique et Intelligence Artificielle* (RIA) du LAAS – CNRS. Une de ces méthodes, le suivi d'un ensemble de points, a été développée initialement dans le groupe, par R. Murrietta Cid, puis améliorée par V. Ayala Ramirez. Nous avons développé les trois autres, suivi de contours, suivi de motifs et suivi de segments, en adaptant leurs descriptions trouvées dans la littérature aux besoins de la robotique mobile. Dans le chapitre suivant, nous ferons une analyse plus détaillée de leurs caractéristiques spécifiques, ce qui permettra de comparer et classer ces méthodes.

2.2 Snakes ou Contours Actifs

Les contours actifs ou “*Snakes*“ [63] ont été intensivement utilisés pour la segmentation et le suivi d'objets. Cette méthode est basée sur la minimisation d'énergie le long d'une courbe quand elle est déformée par des forces internes et externes. Ces forces sont définies afin d'imposer respectivement, des propriétés pour la forme du contour et pour sa position dans l'image.

2.2.1 Formulation classique de l'énergie

Généralement, un contour actif \mathbf{v} est défini par un ensemble de points de contrôle, à partir desquels une courbe continue est interpolée, par des fonctions B-Splines quadratiques ou cubiques. Ensuite, l'énergie totale pour un contour actif \mathbf{v} décrit avec une représentation paramétrique $\mathbf{v} = (x(s), y(s))$ où s désigne l'abscisse curviligne, peut être écrite comme :

$$E_{tot}(\mathbf{v}) = \int E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s)) ds \quad (2.1)$$

où les indices inférieurs dans E , représentent les énergies interne et externe, respectivement.

L'énergie interne est généralement définie par :

$$E_{\text{int}}(\mathbf{v}) = \int_0^1 [\alpha_1(s) \mathbf{v}_s^2 + \alpha_2(s) \mathbf{v}_{ss}^2] ds \quad (2.2)$$

et l'énergie externe par :

$$E_{\text{ext}}(\mathbf{v}) = \int_0^1 P(\mathbf{v}(s)) ds \quad (2.3)$$

où les indices inférieurs sur \mathbf{v} dénotent l'ordre de différentiation, $\alpha_1(s)$ et $\alpha_2(s)$ sont des facteurs de poids donnés aux termes d'élasticité et de lissage respectivement, et P est l'intensité du potentiel externe. Généralement, ce potentiel P est proportionnel au gradient de l'intensité lumineuse dans l'image (figure 2.1), de sorte que le contour actif se colle sur les discontinuités de luminance.

Dans une première étape, le contour actif est définie par une position initiale de la courbe \mathbf{v} , position définie de manière interactive (un opérateur donne par des clics souris par exemple, les points de contrôle de la courbe) ou par une autre fonction visuelle (détection d'objets spécifiques dans l'image : amers visuels, route, obstacle, visage ou main d'une personne...). A partir de cette position initiale, le contour va évolué, soit dans une même image (segmentation d'un objet), soit dans des images successives d'une séquence (suivi d'un objet) : les forces sont appliquées sur les points de contrôle, qui à chaque itération, peuvent se déplacer dans un voisinage donné (typiquement, dans une fenêtre 3x3 centré sur le point). L'évolution du contour actif est déterminée par son équation de mouvement, dérivée à partir de l'équation (2.1).

Cette équation de mouvement peut être résolue par des méthodes variationnelles [13]. Cependant, les approches variationnelles ont souvent des problèmes lors de la phase d'optimisation : mauvaise stabilité numérique, convergence lente, difficulté pour satisfaire des contraintes fortes... Amini et al., dans [4] ont proposé de résoudre ce problème variationnel par une méthode de programmation dynamique, assurant de cette manière la stabilité. La complexité de cette méthode est donnée par $O(nm^{k+1})$, où n est le nombre de points de contrôle (typiquement, plusieurs dizaines de points), m est le nombre de positions possibles pour un point de contrôle (typiquement, 9 positions) et k est le plus haut ordre de dérivation pour le contour (typiquement, 2). Cette méthode a plusieurs avantages :

- elle peut être accélérée en l'appliquant dans un espace multi échelle,
- elle n'est pas sensible à des minima locaux, comme le sont les solutions par approches variationnelles.

Afin de réduire l'espace m , et donc la complexité interne à la méthode, il est possible de simplifier la recherche de positions pour un point de contrôle, aux lignes droites perpendiculaires à la courbe (figure 2.2). Cependant, avec cette restriction, dans le cas du suivi d'objets, il est nécessaire, la plupart du temps, de faire une redistribution des points de contrôle dans la courbe, parce que, sinon après quelques images dans la séquence, ces points deviendront arbitrairement groupés ou séparés sur la courbe.

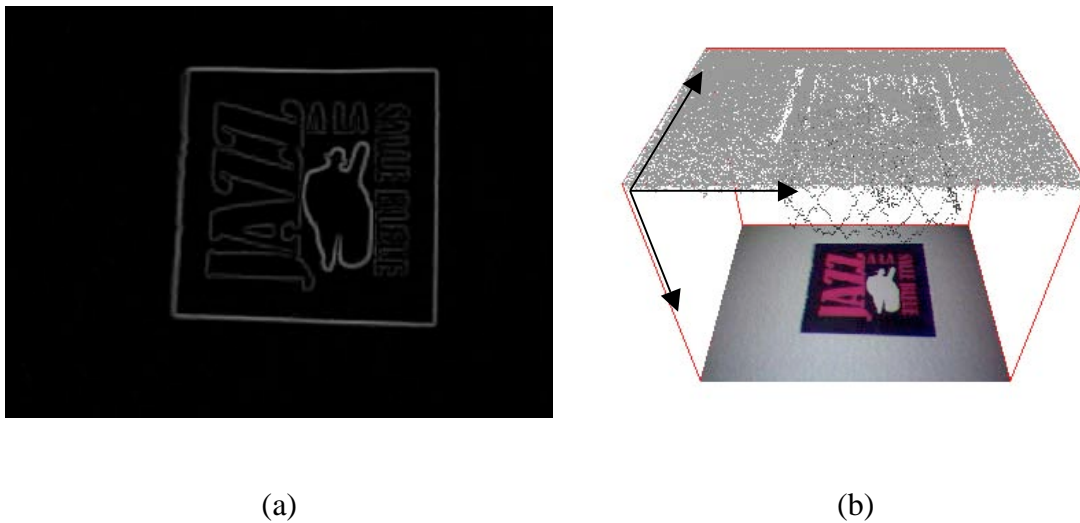


Figure 2.1. Potentiel externe défini à partir du gradient d'intensité lumineuse de l'image, a) image du gradient d'intensité et b) attracteurs (verticalement) qui vont créer le potentiel externe à partir de cette image.

Pour résoudre ce problème, un nouvel échantillonnage de la courbe est nécessaire, afin de redistribuer les points de contrôle avant de traiter l'image suivante, et donc perdre rajouter une étape supplémentaire à l'algorithme.

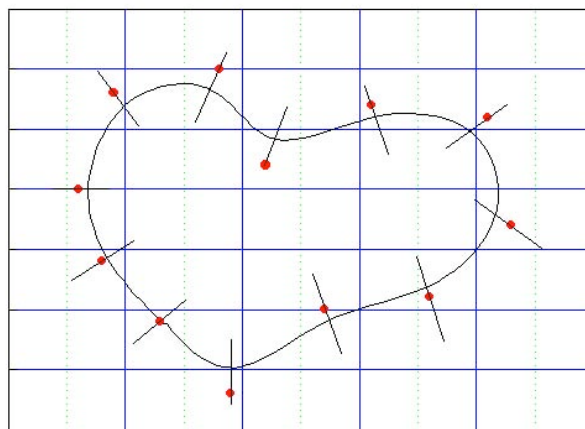


Figure 2.2. Lignes perpendiculaires au contour actif pour la recherche de minima d'énergie.

D'autre part, pour certains objets, comme ceux dont les contours présentent des coins ou des zones de courbure élevée, il est nécessaire d'augmenter le nombre de points de contrôle sur ces zones, afin de les localiser plus finement, mais cela aussi a une répercussion directe sur la complexité. En plus, si le mouvement des points vers ces zones de courbure élevée, n'est pas restreint, la courbe finale peut présenter de boucles.

Dans [75], nous avons proposé de considérer le contour actif comme la surface d'un conducteur électrique, de sorte que, en appliquant une charge électrique totale Q à cette surface, une nouvelle force interne va apparaître. Celle-ci sera une force répulsive dont la magnitude sera en raison de la densité de charge électrique du segment considéré. En conséquence, en divisant cette charge uniformément dans le contour et en

l'appliquant a chaque point de contrôle, cette nouvelle force répulsive redistribuera les points de contrôle sur la courbe avec une plus haute densité des points dans les zones de plus haute courbure convexe [112].

En prenant compte cette nouvelle force, l'énergie interne peut être écrite comme :

$$E_{\text{int}}(\mathbf{v}) = \int \rho_1(s) \mathbf{v}_s^2 + \rho_2(s) \mathbf{v}_{ss}^2 + k \frac{\rho^2}{\mathbf{v}_s^2} ds \quad (2.4)$$

où k est un coefficient constant, et ρ est la densité de charge électrique. Le terme $k \frac{\rho^2}{\mathbf{v}_s^2}$ peut être vu comme une constante, une fois que la charge électrique a été assignée. Selon la loi de Gauss, rappelée dans l'équation (2.5), une fois que l'équilibre est atteint, le flux électrique sur la surface fermée du conducteur, est égal à la charge électrique nette enfermée par la surface, divisée par la constante ϵ_0 (permittivité) ; à l'intérieur du conducteur le champ électrique est exactement zéro [112].

$$\oint \mathbf{E} \cdot d\mathbf{A} = \frac{Q}{\epsilon_0} \quad (2.5)$$

En conséquence, les particules chargées (points de contrôle) sur la surface du conducteur se déplaceront sous l'influence seulement des forces produites par les particules dans leur voisinage, et une fois que l'équilibre est atteint, la densité électrique sera plus haute sur les zones de courbure convexe élevée.

En plus, une fois l'équilibre atteint, la distribution de la densité de charge ne change pas pour un conducteur électrique isolé, quelle que soit sa position dans l'espace. Pourtant, les mouvements de l'objet dans l'espace n'affecteront pas la distribution des charges électriques. La redistribution des points de contrôle par cette force interne peut être utilisée quand les objets sont inconnus ou quand les déformations doivent pousser des points de contrôle dans certaines directions pour mettre à jour la forme du contour.

Nous avons donc développé une méthode de suivi par des snakes, qui utilisent cette stratégie pour redistribuer les points sur le contour. Nous proposons ci-dessous, deux autres variantes de la méthode, dédiées au suivi d'un objet structuré par des snakes, et au suivi d'un objet dans une image couleur.

2.2.2 Définition des modèles déformables

Quand l'objet à suivre, a un modèle connu, en particulier, s'il a une forme structurée, comme un rectangle ou un polygone, nous pouvons imposer des contraintes au modèle du contour actif, afin de le déformer d'une façon appropriée [19]. Des relations entre les points de contrôle peuvent être incorporées sous la forme de termes d'énergie interne.

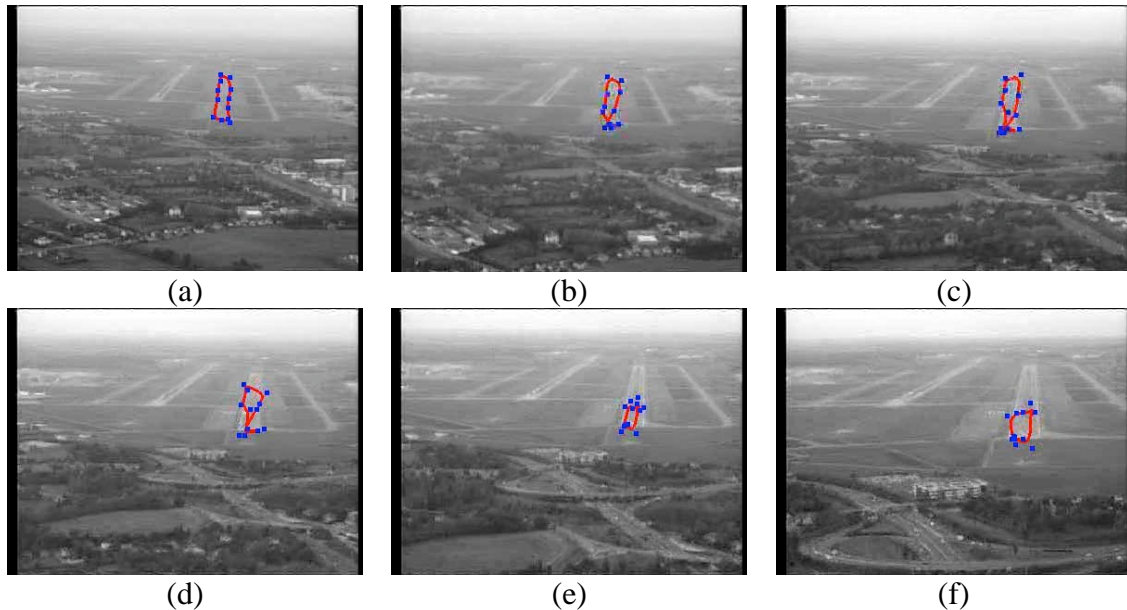


Figure 2.3. Suivi d'une piste d'atterrissage depuis une caméra embarquée sur un avion.

La figure 2.3 montre le suivi d'une piste d'atterrissage par des contours actifs, dans une séquence acquise depuis un avion. Dans les images (a), (b) et (c), le suivi se déroule relativement bien. Cependant, à un moment donné (d), le contour est attiré vers une zone de plus forts gradients ; dans ce cas, cette zone correspond aux marquages blancs inscrits sur la piste, qui servent de référence au pilote de l'avion. De ce fait, dans les images suivantes (e) et (f), le contour est complètement déformé : les gradients en haut de la piste sont trop faibles pour attirer le contour et le remettre sur la piste, si aucune autre contrainte est rajoutée.

Une autre solution pour le suivi d'un objet connu par un modèle consiste à exprimer ce modèle par une courbe paramétrée, puis d'appliquer l'algorithme de CONDENSATION proposé par Isard et Blake [53] ; cette méthode est notamment adaptée pour le suivi d'objets déformables, représentés par un modèle paramétrique. Notons cependant, que si la quantité de paramètres nécessaires pour décrire cette courbe augmente, alors le calcul en temps réel devient difficile.

Afin d'éviter l'utilisation de ces modèles paramétriques complexes, et ainsi assurer le suivi en temps réel, nous avons défini des contraintes locales simples.

Considérons un contour actif \mathbf{v} , défini par des B-splines calculées à partir d'un ensemble ordonné de points de contrôle ; par la suite un **contour actif** peut être défini en utilisant les considérations suivantes :

- a) une séquence de points de contrôle avec différentes multiplicités pour chacun d'entre eux, et
- b) des contraintes de forme entre les points de contrôle pour introduire des segments en des endroits désirées sur la courbe.

Le premier aspect permet d'introduire des discontinuités dans un contour, et le deuxième introduit des restrictions de forme, pour faire apparaître un alignement entre certains points de contrôle successifs. Avant de présenter un résultat de cette méthode, nous explicitons les deux contraintes rajoutées au modèle classique des *snakes*.

Introduction de sommets sur un contour actif

Pour ce qui est de la partie a) de la définition d'un contour actif déformable, il est clair qu'un point de contrôle de multiplicité k produira $k-1$ segments de plus dans la courbe [36]. Sur la figure 2.4, sont montrées, trois courbes B-Splines cubiques avec un point \mathbf{P} de multiplicité 1, 2 et 3, respectivement. Nous pouvons voir dans la figure 2.4c qu'un point de contrôle triple dans une B-Spline cubique produit une discontinuité de la courbe. Ainsi, des points de contrôle multiples permettent d'introduire des coins ou des discontinuités dans la courbe.

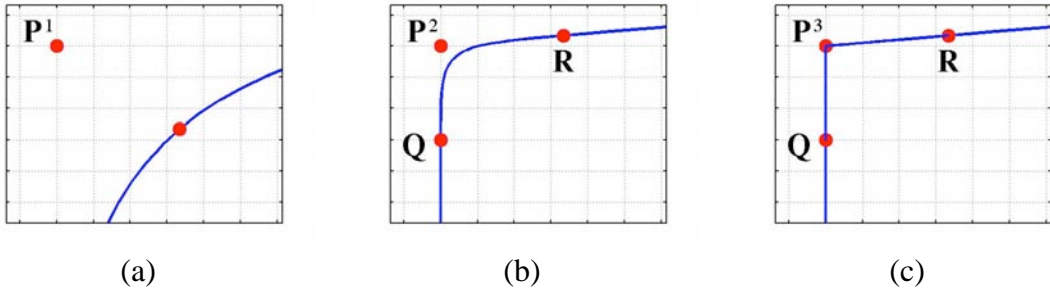


Figure 2.4. B-Spline cubique avec un point de contrôle \mathbf{P} de multiplicité un, deux et trois respectivement.

Soit \mathbf{P} un point de contrôle de multiplicité $k = 3$, tel que celui décrit dans la figure 2.4c. Quand ce point \mathbf{P} est déplacé, les positions des points \mathbf{Q} et \mathbf{R} sont directement affectées, et conséquemment, l'énergie des segments de la courbe, qui contiennent ces trois points (comme début ou fin d'un segment), est également affectée. Pour cette raison et afin d'éviter des multiples calculs entre chaque segment impliqué dans le mouvement d'un point de contrôle de multiplicité plus grand que 1, le calcul de l'énergie pour ces segments doit être traité ensemble.

En résumé, pour un point de contrôle de multiplicité k , nous devons rassembler l'énergie des $k-1$ segments de la courbe directement impliqués, comme cela est décrit dans l'équation suivante :

$$E_{\mathbf{P}_{mult}} = \prod_i^{i+k-1} E(v_i(s)) \quad (2.6)$$

où i est le premier segment qui contient un point de contrôle multiple, et $E(v_i)$ est l'énergie pour le i -ème segment.

Par ailleurs, nous pouvons introduire d'autres restrictions afin d'imposer certains rapports de forme. Sur un point de multiplicité trois, pour favoriser un angle droit sur ce sommet, donc un triangle rectangle sur cette zone du contour, le terme d'énergie des contraintes de forme, pourra être calculé en utilisant des relations dans le triangle \mathbf{PQR} , comme cela est indiqué dans l'équation suivante :

$$E_{\mathbf{P}_{cons}} = c |QR|^2 \left(QP^2 + PR^2 \right) \quad (2.7)$$

où c est un coefficient et QR , QP et PR sont les distances entre les points.

En conclusion, la minimisation de l'énergie pour les $k-1$ segments autour d'un point multiple, permet d'imposer pour un coin dans un contour, une forme désirée, que nous avons donc transformées en une contrainte, qui donnera un terme d'énergie.

Introduction de contraintes de forme sur un contour actif

D'autres contraintes de forme peuvent être facilement prises en compte, en indiquant des zones potentielles de mouvement pour un point de contrôle donné, à partir d'une position initiale de certaines caractéristiques. Par exemple, considérons un contour actif initial où un sous-ensemble de points de contrôle a été caractérisé pour rester alignés ; une zone potentielle de mouvement peut être définie sur ce segment, par la mesure de la différence entre la position actuelle de ce sous-ensemble de points de contrôle, et sa position initiale. Les zones potentielles pour le mouvement des points de contrôle intégrés dans un segment de ligne droite, sont présentées dans la figure 2.5a ; l'énergie à minimiser en fonction de ce potentiel, est calculée comme la distance du point à cette même ligne droite sur le contour actif précédente, en tenant compte des variations possibles de sa pente.

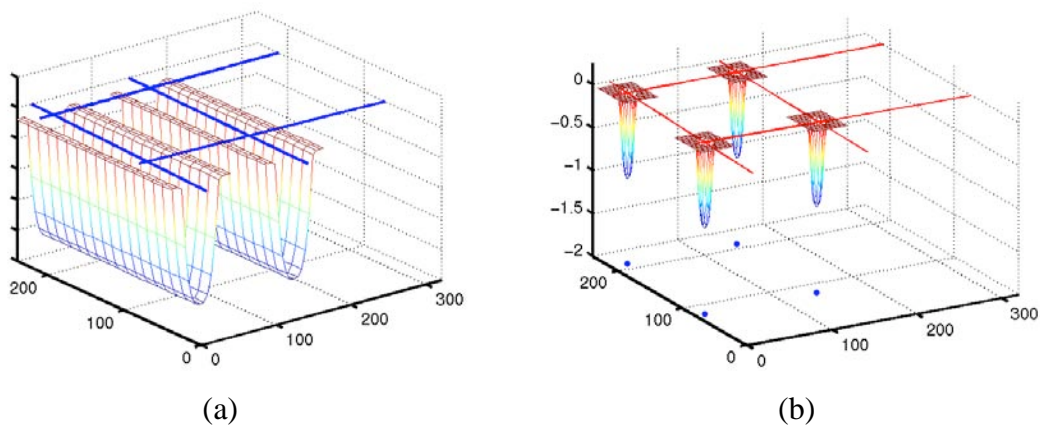


Figure 2.5. Zones potentielles pour la recherche de : (a) lignes droites, et (b) croisement de lignes droites.

Nous avons donc défini des contraintes pour imposer des coins et des segments sur un contour ; mais rien n'impose à ce stade que les segments s'intersectent sur les coins. Pour ce faire, les points multiples peuvent être également contraints à se maintenir à une distance minimale du croisement des lignes droites qui les encadrent, grâce à la zone décrite en figure 2.5b.

Selon la nature et la position d'un segment ou groupe des segments dans le modèle, nous avons aussi essayé d'introduire d'autres potentiels externes pour tenir compte de propriétés dans l'image. Par exemple, une force externe peut attirer un point de contrôle multiple vers un point d'intérêt extrait, par exemple, par un détecteur de Harris [49] appliqué sur une région spécifique. Mais comme cela est bien connu, les détecteurs de coins ne sont pas très précis, et les coins détectés ne font que rajouter des forces qui ne sont pas privilégiées par rapport aux autres.

Bilan, résultats

Une fois que le modèle est défini, en incluant une séquence ordonnée de points de contrôle de multiplicité variable et des contraintes de forme, il est nécessaire d'ajouter un nouveau terme d'énergie à l'équation (2.1). Nous obtenons l'équation suivante :

$$E_{tot}(\mathbf{v}) = \int [E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s)) + E_{cons}(\mathbf{v}(s))] ds \quad (2.8)$$

Le terme E_{cons} , dans (2.8) traduira en énergie, les contraintes de forme décrites dans les points a) et b) de la définition d'un contour actif déformable.

Les modifications introduites de cette façon sur les contours actifs, n'affectent pas beaucoup la vitesse du module qui permet de les suivre, même dans le cas de la programmation dynamique, cela parce que les contraintes introduites sont calculées localement. Les calculs exigés sont très simples, et requièrent les mêmes données que celles également utilisées pour l'évaluation des autres termes d'énergie.

Nous avons aussi essayé d'introduire des contraintes plus globales, par exemple d'imposer le parallélisme entre des segments du contour (pour le suivi d'un quadrangle vu de manière frontale), mais en ce cas, la vérification de telles contraintes introduit des traitements plus lourds incompatibles avec les contraintes temps réel.

L'application de telles contraintes sur la séquence d'atterrissage, ne s'est pas révélée intéressante, du fait de la faible qualité de ces images (faible gradient sur la partie lointaine de la piste). Prenons plutôt comme exemple, celui décrit dans le chapitre précédent, dans le cadre des applications en robotique mobile en intérieur. Il consiste à suivre des objets plans quadrangulaires, pendant le déplacement du robot porteur de la caméra, notamment la navigation d'un robot dans les couloirs. Les portes, affiches et panneaux sont détectés et suivis dans les séquences d'images ; ces objets seront utilisés plus tard en tant qu'amers pour la localisation du robot dans des environnements humains.

Notre modèle pour suivre un quadrangle dans une séquence, est donc défini par un ensemble de points de contrôle ; quatre d'entre eux ont une multiplicité 3 (discontinuité pour un B-Splines cubique). Le nombre de points de contrôle entre ces coins est défini à l'initialisation, selon la distance inter points dans l'image (qui dépendra de la résolution et de la taille de l'amer considéré).

Localement, les points de contrôle de multiplicité égal à un, sont contraints à se déplacer dans une zone potentielle de recherche telle que celle définie précédemment en figure 2.5a. Les points multiples ou coins sont également contraints à se maintenir, dans les zones décrites en figure 2.5b. Les zones potentielles de recherche sont calculées pendant le processus d'initialisation du contour initial, défini sur la première image de la séquence par l'opérateur ou par une autre méthode de reconnaissance automatique. Ces zones sont mises à jour quand l'équilibre est atteint après le suivi de l'objet sur chaque image.

En figure 2.6 est montré le suivi d'une affiche sur un mur. Comme on peut le constater, le suivi de l'objet est fait relativement bien. Cependant, on peut remarquer que tous les coins ne sont pas extraits sur toutes les images ; notamment dans l'image(b), le coin inférieur droit et dans l'image (c), le coin supérieur droit. Cela s'explique car les contraintes ne sont pas forcées. En effet, ces contraintes sont

exprimées sous la forme de forces complémentaires ; la somme totale des forces donnera la meilleure position pour un point de contrôle.

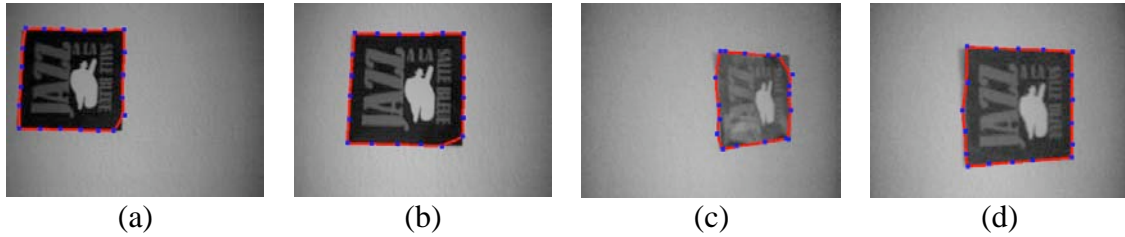


Figure 2.6. Suivi avec modèles de contours actifs déformables d'une affiche sur un mur.

2.2.3 Snakes Couleur

Des contours actifs dans des images couleur, définis en exploitant réellement l'information des couleurs [93] ont été utilisés dans plusieurs études. Par exemple dans [46], afin d'obtenir des contours actifs invariants en fonction de la couleur, les trois caractéristiques suivantes dérivées de l'espace RGB (par ses sigles en anglais, *Red*, *Green* et *Blue*) ont été analysées :

a) l'intensité

$$I = R + G + B, \quad (2.9)$$

b) les couleurs normalisés, et

$$r = \frac{R}{R + G + B},$$

$$g = \frac{G}{R + G + B}, \quad (2.10)$$

$$b = \frac{B}{R + G + B}$$

c) la teinte ou *hue*,

$$H = \arctan \frac{\sqrt{3}(G - B)}{(R - G) + (R - B)} \quad (2.11)$$

Ainsi, en étudiant et comparant différents gradients couleur pour ces caractéristiques, ils montrent que c'est le gradient appliqué à la composante Teinte et ceux appliqués aux couleurs normalisées qui ont la meilleure invariance aux changements dans les couleurs, dus aux variations d'illumination ou de point de vue. Cependant, la validation de cette étude n'a été faite que sur des objets artificiels avec des couleurs mates et brillantes (style jeux de cube pour enfants), sous des conditions contrôlées d'illumination. Comme nous allons le montrer, ces caractéristiques ne sont pas suffisantes pour des images complexes.

Dans [56], pour le suivi de régions dans des images couleur, la notion de région active a été définie. L'énergie externe du contour actif est remplacée par une force

émanant de la région à suivre, force définie comme une fonction de correspondance multi canal (couleur). Comme le suivi doit être effectué à la fréquence vidéo, les auteurs ont choisi d'utiliser les couleurs normalisées. Cependant, ils bénéficient aussi de conditions contrôlées d'illumination et le suivi s'applique à des pièces de couleur mate, d'un bras robotique.

Dans des environnements naturels ou complexes, il est très difficile d'avoir des régions avec de couleurs mates, ou avec une texture uniforme, comme cela a été supposé dans les travaux évoqués ci-dessus. Il est presque impossible d'avoir des conditions d'illumination contrôlées. De ce fait, il n'est pas facile de définir une bonne représentation colorimétrique afin de décrire des régions. Nous pouvons voir dans la figure 2.7, que sur de telles images, ni le gradient d'intensité, ni les gradients couleurs ne sont suffisamment robustes pour garantir une bonne stabilité et une convergence rapide du suivi des caractéristiques désirées, la route dans cet exemple.

Nous avons essayé d'analyser et de trouver un espace couleur avec les caractéristiques désirées, qui puisse être utilisé pour filtrer, d'une manière facile et robuste, une région spécifique grâce à ces caractéristiques colorimétriques en temps réel. Malheureusement, définir ou caractériser une zone par sa couleur ou sa texture comme cela a été fait dans [55], est très difficile sans le contrôle des conditions d'illumination.

Nous avons analysé les espaces couleur suivants : l'espace RGB , l'espace de couleurs normalisés et l'espace $I_1I_2I_3$ défini par Ohta et al. [88], et décrit comme :

$$\begin{aligned}
 I_1 &= \frac{R + G + B}{3} \\
 I_2 &= R - B \\
 I_3 &= G - \frac{R + B}{2}
 \end{aligned}
 \tag{2.12}$$

Nous avons comparé ces espaces avec les résultats obtenus pour les images d'intensité.

Dans la figure 2.7, nous montrons les résultats obtenus ; comme cela apparaît (quand les contours peuvent être distingués), aucun de ces espaces ne donne une bonne segmentation des régions seulement à partir des informations couleur. Dans [24], le problème de la segmentation couleur, est analysé : il est montré que les espaces couleur ne sont pas suffisamment discriminants pour avoir une bonne segmentation des zones de couleur ou texture homogène.

Dans le groupe RIA, R.Murrietta Cid, puis Gabriel Avina Cervantès [5], ont développé une méthode de segmentation de régions capable d'extraire des régions homogènes avec une bonne robustesse ; cette méthode procède en trois étapes

- Une segmentation de l'image couleur, par *clustering*, ou classification non supervisée : cette méthode se fonde sur l'espace $I_1I_2I_3$ défini par Ohta. Le nombre de classes est choisi pour produire une sur-segmentation (typiquement 5 classes par composante colorimétrique, donc 125 classes au total) ;

- Une classification des régions extraites, en utilisant une base d'apprentissage construite en mode supervisé, hors ligne ; les régions sont caractérisées par un vecteur d'attributs couleur/texture dans \mathbb{R}^{10} ; les classes apprises correspondent à des objets typiques qui se rencontrent en milieu naturel (arbre, ciel, rocher, goudron, terre, herbe...).
- Une fusion des régions connexes de même classe.

Cette méthode permet d'extraire des régions « route » avec un taux de réussite supérieur à 90%. D'autres critères sont en cours d'étude pour augmenter encore ce score.

Nous pouvons donc considérer que notre cible (la route en ce cas), peut être identifiée et caractérisée par cet algorithme. Il est alors possible de trouver les paramètres nécessaires pour effectuer le suivi de cette cible dans un espace de caractéristiques bien définies. Dans ce contexte, nous avons proposé une méthode de suivi, similaire à celle décrite dans [56] : nous pouvons considérer qu'il s'agit d'une méthode de suivi d'un contour actif fermé, qui définit une région uniforme segmentée par ailleurs.

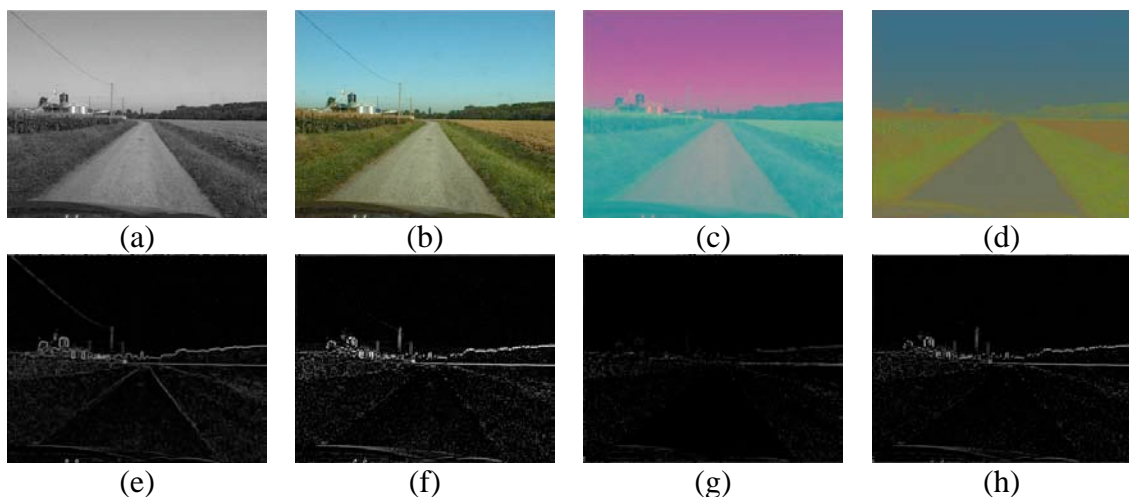


Figure 2.7. Espaces couleur et gradients: a) image d'intensité, b) image couleur (RGB), c) espace d'Ohta, d) espace de couleurs normalisées, e), f), g), et h) sont les gradients pour les images au-dessus respectivement.

Il s'agit plus précisément d'une méthode qui exploite une collaboration entre deux fonctions visuelles, traitant respectivement d'interprétation sur une image initiale et de suivi sur une séquence suivante. La méthode globale impliquant des interactions entre deux modules, sera décrite avec plus de détail dans le chapitre 5 . Ici nous analysons la représentation des contours actifs définis par des régions.

Considérons la sortie de la méthode de segmentation et caractérisation en régions : cette sortie nous donne les caractéristiques statistiques de la région à suivre. A partir de ces caractéristiques, on peut analyser quel espace couleur est le meilleur pour faire le suivi de cette région spécifique. Par exemple pour le suivi de la route décrit précédemment, il est possible de voir dans la figure 2.8, que cette zone est mieux définie dans l'espace de couleurs normalisées. Même, si la région *route* n'est pas complètement isolée du reste des régions, les résultats sont nettement meilleurs que dans les analyses précédentes.

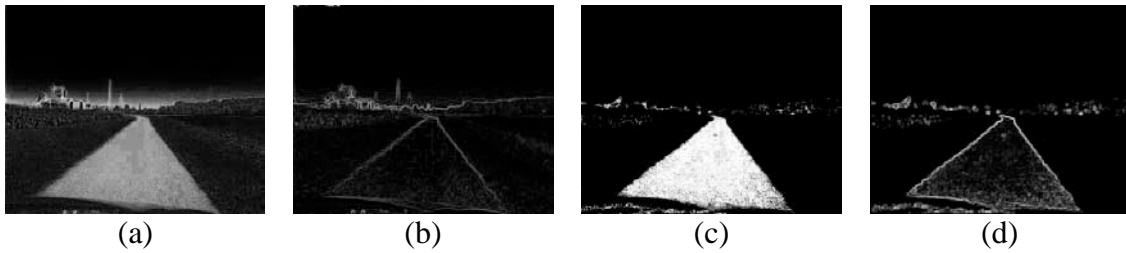


Figure 2.8. Caractérisation et détection de régions à partir des résultats de la méthode de segmentation : a) caractérisation dans l'espace de Ohta b) gradient d'intensité dans cet espace, c) caractérisation dans l'espace de couleurs normalisés, d) gradient d'intensité dans cet espace.

L'espace dans lequel chaque cible ou région pourrait être suivie, dépend autant de la nature de la cible que de la méthode utilisée pour sa détection. Bien sûr, le suivi sera possible en temps réel tant qu'il existe une transformation locale entre l'espace initial et celui où la méthode sera appliquée.

Citons un point important, qui sera décrit dans le chapitre 5 , mais qui vaut la peine d'être mentionné ici. La méthode de segmentation et caractérisation de la cible est effectuée en continue, mais du fait de sa complexité, à une fréquence relativement faible, vis-à-vis du suivi (typiquement, un facteur 20 entre les deux fréquences). Il sera donc possible toutes les 20 itérations du suivi, d'adapter les caractéristiques de la cible et de prendre ainsi en compte des variations sur les conditions d'illumination.

Dans la figure 2.9 nous montrons le suivi de route pour un véhicule se déplaçant en rase campagne. Les images présentent de légers changements de texture et d'illumination. Des résultats similaires, mais moins stables, ont été obtenus sur des chemins de terre.

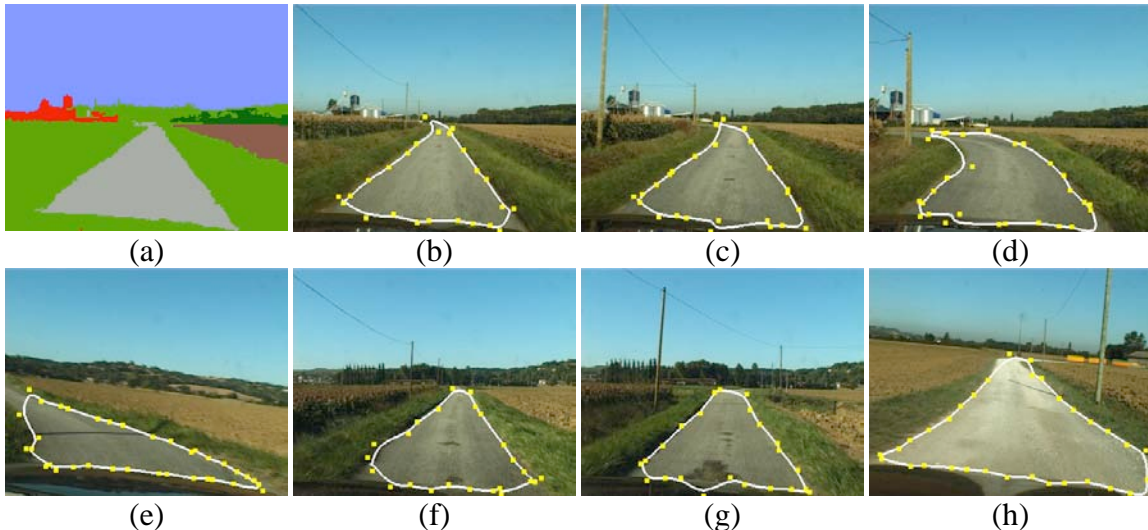


Figure 2.9. Résultats du suivi d'une route : a) exemple d'image avec les zones segmentés utilisé pour la caractérisation de paramètres pour le suivi de contours, et de b) à h) suivi de la région par des contours actifs.

2.3 Suivi avec corrélation 1D

La méthode de suivi d'objets par suivi de lignes, a été une des premières méthodes proposées dans la littérature. Initialement, pour chaque image d'une séquence, les lignes (segments de droite ou d'ellipse) étaient d'abord extraites, puis appariées avec leurs positions prédites fournies par le traitement de l'image précédente. Il est admis à présent que ces méthodes ne sont pas suffisamment robustes, du fait des erreurs dans la phase d'extraction des lignes. Par exemple l'extraction de segments de droite peut donner des résultats instables d'une image à l'autre, du fait de la grande sensibilité des opérateurs qui extraient les gradients de luminance, puis des méthodes d'amincissement et d'extraction des contours et enfin, de leur approximation polygonale.

Cependant de nouvelles approches, ou quelques changements et améliorations des approches existantes, font du suivi de lignes, l'une des méthodes les plus utilisées dans la communauté.

L'idée principale dans la plupart des ces approches, est d'éviter l'extraction explicite des lignes sur l'image traitée. En supposant qu'il existe (1) une prédiction sur la position de la ligne recherchée et (2) un modèle de cette ligne, la méthode consiste à trouver dans une direction et à une distance données de la ligne prédite, des points qui pourraient appartenir à une ligne définie par un modèle a priori, par exemple (figure 2.10),

- une marche clair-sombre ou sombre-clair, comme représentée en figure 2.10 à droite
- un pic clair sur fond sombre,
- un creux sombre sur fond clair
- un point de contour sur le gradient de l'image
- ou un point de Harris.

C'est avec ces points, qu'il sera possible de reconstruire la ligne recherchée à partir d'une méthode robuste d'ajustement de modèles (RANSAC par exemple). Cette ligne, éventuellement mise à jour par une méthode d'estimation de la dynamique de la cible, deviendra prédiction pour l'image suivante, et le suivi pourra ainsi se faire dans une séquence d'images. Cette idée a été utilisée pour suivre des formes très différentes comme par exemple : des lignes droites, des arc d'ellipses [109], ou des structures plus complexes comme dans [39].

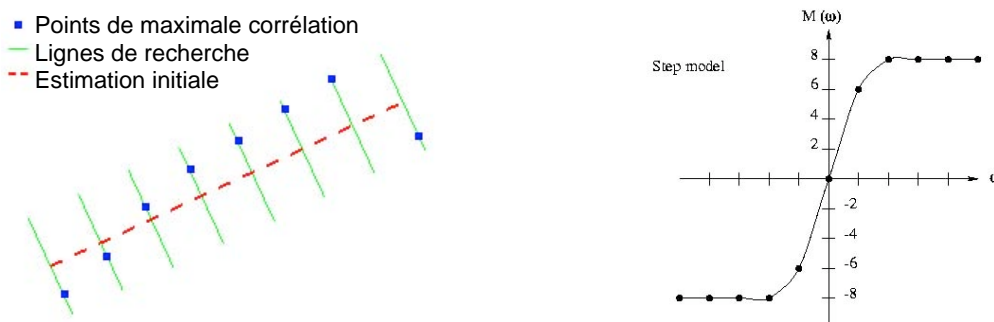


Figure 2.10. Principe de la corrélation 1D. **A gauche**, la ligne rouge en trait pointillé donne la position prédite de la ligne recherchée ; la corrélation s'effectue le long des segments verts de direction orthogonale à cette ligne prédite ; les points bleus sont les maximaux de corrélation dans ces directions. **A droite**, le modèle d'une discontinuité foncé-clair de luminance.

Nous illustrons en figure 2.10 le suivi par corrélation 1D d'un segment de droite. Etant donnée une ligne droite initiale prédite l , on va *recaler* ses paramètres aux

données de l'image courante en fonction d'un modèle voulu de cette ligne. A partir de cette ligne l sont définies m directions dans lesquelles on va chercher une corrélation maximale ; dans notre cas se sont des lignes perpendiculaires à la droite initiale. La corrélation cherchée sera définie afin de trouver soit un point de contour, ou soit une discontinuité de l'intensité lumineuse.

Une fois trouvés tous les points avec une corrélation maximale, il est nécessaire d'adapter les paramètres de la droite aux points corrélés. Comme il est décrit dans [65], l'ajustement des paramètres une tâche très difficile à effectuer, quand il existe des grosses erreurs, qui ne sont pas uniquement des erreurs dûs au bruit dans les images, mais qui peuvent provenir d'erreurs implicites aux méthodes de corrélation. Dans ces cas, par exemple l'ajustement des paramètres par simple utilisation des moindres carrés ou MLE (Maximum Likelihood Estimation) n'est pas très satisfaisante [79], [102]. D'une part l'utilisation de MLE n'est pas très valable du fait principalement que les populations qu'on obtient sont rarement suffisantes pour l'utilisation des méthodes statistiques, et d'autre part les gros erreurs produites par la corrélation peuvent avoir des effets catastrophiques sur l'estimation (figure 2.11).

Afin d'éviter ces erreurs d'estimation en présence de points aberrants ou *outliers*, plusieurs méthodes sont apparues dans la littérature depuis une dizaine d'années ; nous utilisons la méthode de RANSAC (RANDOM SAMPLE CONSENSUS), décrite dans [44]. Cette méthode nous permet de rejeter les grosse erreurs, et de donner une estimation des paramètres de la courbe à extraire ou à suivre avec au moins 50% de tous les points corrélés.

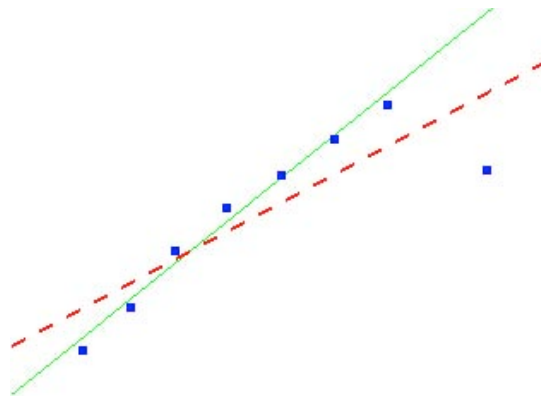


Figure 2.11. Ajustement de paramètres une ligne droite avec des grosses erreurs par moindres carrés.

Pour résumer, dans notre implémentation, la méthode nécessite une prédiction donnée de la position de la droite l^i , avec $i = 1 \dots n$, à chaque instant t de la séquence. m points seront échantillonnés régulièrement sur cette droite. Ensuite pour chaque point j sur la droite, avec $j = 1 \dots m$, il sera cherché sur la droite perpendiculaire à l^i , quel point a le plus haute valeur de corrélation vis-à-vis d'un modèle donné (modèle de marche dans notre cas). Et finalement c'est avec la méthode de RANSAC qu'on va ajuster les paramètres de chaque droite à ces points. Si, parmi les m points échantillonnés sur la droite prédite, il y en a au moins 50%, alignés sur la droite recalée, le suivi aura réussi.



Figure 2.12. Suivi d'une piste par corrélation 1D.

Les positions initiales des lignes ($l^i_0, i=1\dots n$), sont donnés, soit par une projection 3D de segments dans l'image (par exemple, mur-sol, mur-toit), soit par les contours d'un objet reconnu dans l'image.

Nous avons implémenté cette méthode principalement pour le suivi de n lignes droites indépendantes, correspondant à des arêtes dans la scène 3D (transition mur-plafond ou mur-sol, arêtes verticales ...). La figure 2.13, présente le suivi d'une ligne droite par la méthode de corrélation 1D, dans ce cas la ligne est l'arête entre le sol et le mur.

Cette méthode a été utilisée aussi pour suivre des objets structurés comme des quadrangles, typiquement les affiches murales. Mais, dans notre implémentation nous ne vérifions pas des restrictions entre plusieurs lignes telles que le parallélisme, la convergence ou les points à l'infini. Par exemple, figure 2.12, nous avons appliqué cette méthode sur la séquence d'atterrissage ; le suivi est initialisé de manière interactive sur la première image. La piste est suivie en considérant qu'elle est plus claire dans l'image d'intensité que les régions environnantes (un modèle couleur des frontières de la piste eut été probablement meilleur) ; nous pouvons constater des instabilités, qui conduisent dans la suite de cette séquence, à des erreurs (croisement des côtés du quadrangle)

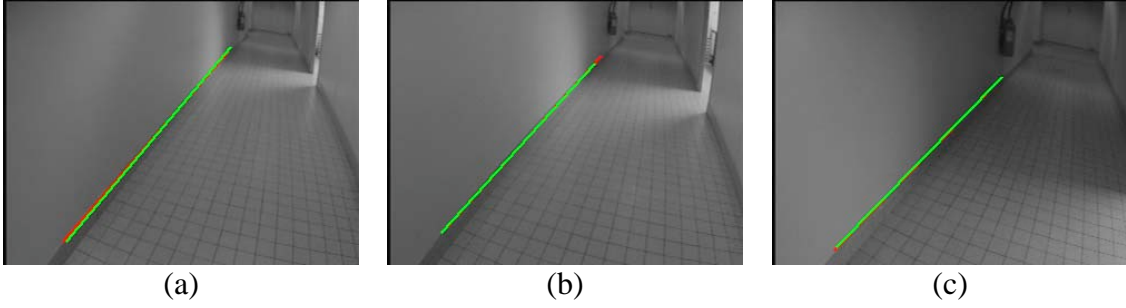


Figure 2.13. Suivi de lignes par corrélation 1D.

2.4 Ensemble des points

Cette méthode de suivi a été présentée dans [52], [81], [7]. Le suivi est fait en utilisant une comparaison entre deux groupes de points,

- d'une part les points extraits dans une région d'intérêt dans l'image courante : points de discontinuités, détectés par une méthode classique de contours : Canny, Deriche, ... ou points d'intérêt extraits par détecteurs comme Susan, Harris, ...,
- et de l'autre part, les points contenus du modèle de la cible, qui proviennent généralement de l'itération précédente du suivi.

La distance de Hausdorff est utilisée pour mesurer la similitude entre ces groupes de points. Dans d'autres approches il a été proposé de suivre seulement des points, sans appliquer des restrictions à leur positions relatives, en utilisant une fonction de corrélation indépendante pour plusieurs fenêtres (typiquement 10x10), sélectionnés au tour de points d'intérêt.

Une distance partielle de Hausdorff a été utilisée comme mesure de ressemblance entre le modèle de la cible et sa position prédit dans l'image. La distance d'Hausdorff pour deux groupes de points \mathbf{P} et \mathbf{Q} est définie comme :

$$H(\mathbf{P}, \mathbf{Q}) = \max(h(\mathbf{P}, \mathbf{Q}), h(\mathbf{Q}, \mathbf{P})) \quad (2.13)$$

où

$$h_k = K^{th} \min_{\mathbf{q} \in \mathbf{Q}} \|\mathbf{p} - \mathbf{q}\| \quad (2.14)$$

et où $\|\mathbf{p} - \mathbf{q}\|$ est la distance entre deux points \mathbf{p} et \mathbf{q} . $K^{th} \min_{\mathbf{p} \in \mathbf{P}} f(\mathbf{p})$ dénotes le K^{th} valeur de $f(\mathbf{p})$ dans le group \mathbf{P} .

La fonction $h(\mathbf{P}, \mathbf{Q})$ (distance du groupe \mathbf{P} au groupe \mathbf{Q}) est une mesure du degré auquel chaque point dans \mathbf{P} est proche d'un point dans \mathbf{Q} . La distance de Hausdorff est la distance maximale entre $h(\mathbf{P}, \mathbf{Q})$ et $h(\mathbf{Q}, \mathbf{P})$; elle donne la distance entre le pire appariement de points dans le deux groupes, en rejetant les K pires appariements, qui sont considérées comme des erreurs.

A l'instant $t+1$ de la séquence, la première tâche, consiste à rechercher la position du modèle M_t dans l'image suivante I_{t+1} , autour de la position prédite. La valeur minimale de la distance unidirectionnelle partielle du modèle à l'image, $h_{k1}(M_t, I_{t+1})$ identifie la meilleure position de M_t dans I_{t+1} , sous l'action d'un groupe de translations \mathbf{G} . Pour optimiser le temps d'exécution de la méthode, la recherche de la cible est

arrêtée quand, la première translation g , tel que la distance associée $h_{k1}(M_t, I_{t+1})$ n'est pas plus grande qu'un seuil donné \square

La stratégie considérée, pour la recherche de la cible est une trajectoire en spirale autour de la position prédite.

Une fois qu'on a trouvé la position de la cible M_t dans l'image I_{t+1} de la séquence, la deuxième tâche consiste à actualiser le modèle de la cible M_t : le nouveau modèle M_{t+1} , est construit, en déterminant quels pixels de l'image I_{t+1} , font partie de la cible. Le modèle est actualisé en utilisant la distance unidirectionnelle partielle de l'image au modèle comme critère pour sélectionner le sous-groupe de points image I_{t+1} qui appartient au M_{t+1} .

Le nouveau critère est défini comme :

$$M_{t+1} = \{g \square I_{t+1} \mid h_{k2}(I_{t+1}, g(M_t)) < \square\} \quad (2.15)$$

où $g(M_t)$ est le modèle au temps t sous l'action de la translation g , et \square contrôle la capacité qu'aura la méthode afin de suivre les objets qui changent de forme.

Afin de permettre des changements d'échelle, le modèle de la cible est grossi quand il y a un nombre significatif de pixels non nuls, proches de la boîte englobante de l'ensemble de points ; elle est réduite dans le cas opposé. Signalons que cette stratégie pour les changements d'échelle est très heuristique et en pratique, est souvent mise en échec.

Le modèle M_t peut être filtré en appliquant des restrictions de forme à la cible, au moyen d'opérateurs morphologiques (par exemple, cibles de forme elliptique pour suivre des visages).

Finalement, pour résumer, à partir d'une position initiale de la cible, donnée par un module de détection, la méthode suivra progressivement la position de la cible et actualisera son modèle, en utilisant les deux règles suivantes :

$$k1 > fM \mid h_{k1}(M_t, I_{t+1}) < \square \quad (2.16)$$

et

$$k2 > fI \mid h_{k2}(I_{t+1}, g(M_t)) < \square \quad (2.17)$$

dans laquelle fM est une fraction du nombre total des points du modèle M_t qu'il faudra apparier dans l'image courante, et fI est une fraction des points de la zone image apparée I_{t+1} , qu'il faudra retrouver dans le modèle soumis à une translation $g(M_t)$.

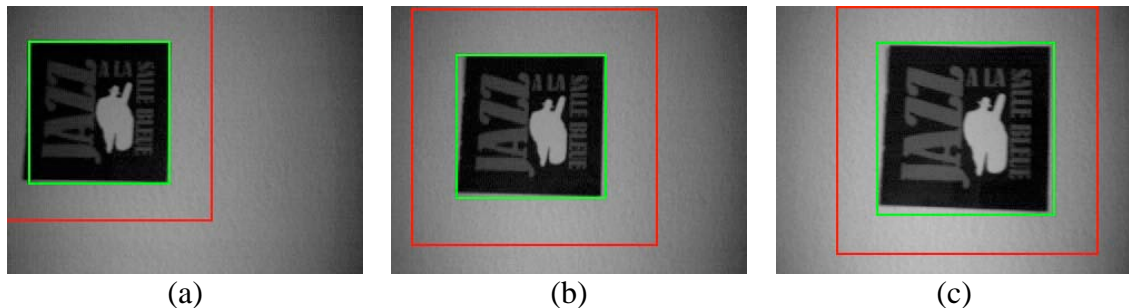


Figure 2.14. Suivi d'un ensemble de points en exploitant la distance de Hausdorff.

Dans la **figure 2.14**, nous montrons le suivi d'un poster sur un mur, le rectangle intérieur représente la zone de la cible, et l'extérieur la zone de recherche.

2.5 Suivi par différences des Motifs

Les méthodes décrites dans cette section sont basées sur la différence des motifs (ou modèles) entre deux régions. Les méthodes basées sur l'appariement de régions, ont été utilisées dans des contextes très différents tels que : l'appariement en stéréo, le calcul de flux optique ou encore, l'analyse de mouvement.

L'idée principale est de mesurer à un instant donnée t , la différence entre un motif pré déterminé (de référence) et une position possible de ce motif dans l'image : position issue d'une prédiction, ou issue d'un module de recherche exhaustive dans l'image. Typiquement, la *somme de différences carrées* (ou *SSD* en anglais) ou la *somme des valeurs absolues des différences* (ou *SAD* en anglais) sont des mesures de **corrélation** rapides à calculer, qui permettent de statuer sur la ressemblance des motifs : soit appairer la position prédite avec le motif de référence, soit refuser cette prédiction et en essayer une autre. Ces scores sont rapides à calculer, mais très sensibles aux conditions d'illumination et aux occultations. JB.Hayet [51] a défini une mesure partielle de corrélation, qui permet de rendre ces techniques plus robustes ; dans son cas, il utilise un *coefficient de corrélation centrée et normalisée* (ou *ZNCC* en anglais).

La plupart des méthodes utilisent une mesure de corrélation entre les deux régions, telle que $L_{2\text{norme}}$. Dans [85], différentes mesures de confiance ont été étudiées pour obtenir une meilleure estimation de la corrélation entre des régions ; une mesure à partir d'une distribution gaussienne associée au motif a aussi été proposée.

Malheureusement, le plupart de ces mesures de corrélation sont très coûteuses en temps de calcul et donc pas très efficaces pour leur utilisation en temps réel.

Dans les sections suivantes, nous nous focaliserons seulement sur les approches de suivi basées sur les différences de motifs qui proposent d'estimer les variations de pose du motif, et plus précisément sur les méthodes qui ont montré une très bonne efficacité et qui de ce fait, peuvent être implémentées en temps réel.

2.5.1 Approche de Hager et Belhumeur.

Hager et Belhumeur, dans [48], proposent de résoudre le problème du suivi d'objets avec les différences de motifs, par la connaissance a priori du modèle paramétrique de mouvement et l'estimation (hors ligne) d'une matrice *Jacobienne* entre le modèle de mouvement choisi et le motif de référence. Même si nous ne l'avons pas adoptée, nous considérons que cette approche est la base des nouvelles approches plus performantes ; c'est pourquoi nous allons la décrire tout de suite, en commençant par quelques définitions.

Soit $I(\mathbf{x}, t)$, l'intensité lumineuse de la position $\mathbf{x} = (x, y)$, d'une image au temps t , et soit $\mathbf{R} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, l'ensemble de N positions dans l'image, lesquelles définissent la région cible. Subséquemment, soit $\mathbf{I}(\mathbf{R}, t) = (I(\mathbf{x}_1, t), I(\mathbf{x}_2, t), \dots, I(\mathbf{x}_N, t))$ le vecteur des intensités de la région \mathbf{R} au temps t . Finalement, prenons au temps t_0 , tel que $t = 0$, le *motif de référence* $\mathbf{I}(\mathbf{R}, t_0)$.

Le mouvement du motif de référence au temps t , produit des déformations, rotations et/ou translations de ce même motif dans les images suivantes. Considérons que ce mouvement peut être défini par une fonction paramétrique $\mathbf{f}(\mathbf{x}; \boldsymbol{\mu})$, appelée

modèle de mouvement et paramétrée par un vecteur $\boldsymbol{\mu}$, tel que $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)$, où la dimension n du vecteur $\boldsymbol{\mu}$ est le nombre des paramètres pour le modèle de mouvement.

Prenons comme condition initiale, que au temps t_0 , $\boldsymbol{\mu}_0$ est le vecteur des paramètres tel que $\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}_0) = \mathbf{x}$, c'est-à-dire les paramètres pour que \mathbf{f} soit la fonction identité.

Donc, le suivi de l'objet dans une séquence d'images peut être considéré comme le fait de déterminer le vecteur $\boldsymbol{\mu}$ à chaque instant $t > t_0$, de la séquence. Il est nécessaire de considérer, que \mathbf{f} est différentiable, autant pour \mathbf{x} que pour $\boldsymbol{\mu}$.

En notant $\boldsymbol{\mu}^*(t)$ et $\boldsymbol{\mu}(t)$ la valeur réelle et l'estimée du mouvement, respectivement, et en considérant qu'il n'y pas de changements d'illumination dans la séquence d'images, on a:

$$I(\mathbf{x}, t) = I(\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}(t))) \quad \mathbf{x} \in \mathbf{R} \quad (2.18)$$

Pour $N > n$ (N : nombre de points sur le motif, n : nombre de paramètres pour définir le mouvement), le vecteur des paramètres du mouvement $\boldsymbol{\mu}$ peut être estimé à l'instant t en minimisant la *fonction objectif* suivante :

$$O(\boldsymbol{\mu}) = \int_{\mathbf{x} \in \mathbf{R}} (I(\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}), t) - I(\mathbf{x}, t_0))^2 \quad (2.19)$$

En considérant que, au temps $t > t_0$ le vecteur des paramètres de mouvement peut être rapproché par $\boldsymbol{\mu}(t + \Delta t) = \boldsymbol{\mu}(t) + \Delta \boldsymbol{\mu}$, Δ étant un très petit (presque infinitésimal) incrément du temps t , et en écrivant le vecteur image de la région cible comme:

$$\mathbf{I}(\boldsymbol{\mu}, t) = \begin{bmatrix} I(\mathbf{f}(\mathbf{x}_1; \boldsymbol{\mu}(t)), t) \\ I(\mathbf{f}(\mathbf{x}_2; \boldsymbol{\mu}(t)), t) \\ \dots \\ I(\mathbf{f}(\mathbf{x}_N; \boldsymbol{\mu}(t)), t) \end{bmatrix} \quad (2.20)$$

nous pouvons écrire la fonction objectif O (2.19) en termes de $\Delta \boldsymbol{\mu}$, pour obtenir l'équation suivante :

$$O(\boldsymbol{\mu} + \Delta \boldsymbol{\mu}) = \|\mathbf{I}(\boldsymbol{\mu}(t) + \Delta \boldsymbol{\mu}, t + \Delta t) - I(\boldsymbol{\mu}_0, t_0)\|^2 \quad (2.21)$$

Comme il est décrit dans [48], quand les composantes de $\Delta \boldsymbol{\mu}$ sont petites, il est possible de linéariser le problème en faisant un développement de $\mathbf{I}(\boldsymbol{\mu}(t) + \Delta \boldsymbol{\mu}, t + \Delta t)$ en séries de Taylor autour de $\boldsymbol{\mu}$ et par la suite, nous obtenons l'équation suivante :

$$\mathbf{I}(\boldsymbol{\mu} + \Delta \boldsymbol{\mu}, t + \Delta t) \approx \mathbf{I}(\boldsymbol{\mu}, t) + \mathbf{M}(\boldsymbol{\mu}, t) \Delta \boldsymbol{\mu} + \Delta \mathbf{I}_t(\boldsymbol{\mu}, t) + h.o.t. \quad (2.22)$$

où \mathbf{M} est la matrice *Jacobienne* de \mathbf{I} par rapport au $\boldsymbol{\mu}$, et *h.o.t.* dénote les termes de plus haut ordre. Nous avons écrit par simplicité $\boldsymbol{\mu}(t) = \boldsymbol{\mu}$.

Il est possible aussi, de rapprocher \mathbf{I}_t dans (2.22) par :

$$\Delta \mathbf{I}_t(\boldsymbol{\mu}, t) \approx \mathbf{I}(\boldsymbol{\mu}, t + \Delta t) - \mathbf{I}(\boldsymbol{\mu}, t) \quad (2.23)$$

et en substituant (2.23) et (2.22) dans (2.21) et en ignorant les termes de plus haut ordre, on peut résoudre l'ensemble des équations pour $\Delta O = 0$ d'où on obtient la solution suivante, en prenant la pseudo-inverse de la matrice \mathbf{M} :

$$\Delta \mathbf{I} = \mathbf{I}(\mathbf{M}^t \mathbf{M})^{-1} \mathbf{M}^t [\mathbf{I}(\Delta, t + \Delta) - \mathbf{I}(\Delta_0, t_0)] \quad (2.24)$$

Nous définissons le vecteur de différences d'intensité comme :

$$\Delta \mathbf{i} = \mathbf{I}(\Delta, t + \Delta) - \mathbf{I}(\Delta_0, t_0) \quad (2.25)$$

Un point important à prendre en compte, c'est que dans (2.26), le vecteur défini de différences d'intensité $\Delta \mathbf{i}$, est la variation de l'intensité de la cible entre la région de référence et la région à l'instant $t + \Delta$, avec un mouvement $\boldsymbol{\mu}$ au temps précédent t (figure 2.15), d'où la différence entre \mathbf{i} et \mathbf{I} .

En substituant (2.25) dans (2.24), le vecteur de paramètres de mouvement reste comme :

$$\Delta \mathbf{I}(t + \Delta) = \Delta \mathbf{I}(t) \mathbf{I}(\mathbf{M}^t \mathbf{M})^{-1} \mathbf{M}^t \Delta \mathbf{i} \quad (2.26)$$

L'équation (2.26) établit que, le vecteur de paramètres de mouvement $\boldsymbol{\mu}$ au temps $t + \Delta$ peut être calculé à partir de données au temps t , seulement si Δ est suffisamment petit pour assurer de petites variations entre les images successives et donc assurer que *h.o.t.* soient effectivement négligeables. Par conséquence, moyennant ces hypothèses, on peut calculer les paramètres de mouvement $\Delta \boldsymbol{\mu}$.

Dans nos cas, la valeur de Δ est restreint par la fréquence vidéo de la caméra. En conséquence, Δ désignera, à partir de ce moment, le temps entre deux images successives de la séquence.

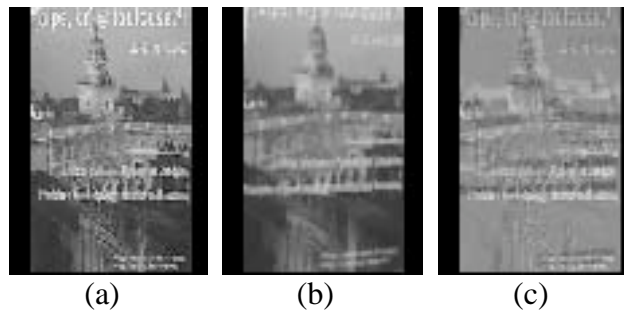


Figure 2.15. Différences de régions : (a) motif de référence, (b) variation de la région (c) image des différences

Hager et Belhumeur [48], proposent différents modèles linéaires pour réduire la complexité du calcul de la matrice *Jacobienne*, comme des translations et rotations pures, des changements d'échelle, ainsi qu'un modèle non linéaire. Dans [22] Buenaposada et Baumela proposent d'utiliser un modèle de mouvement projectif et une procédure pour calculer la position et l'orientation de la région suivie dans chacune des images, en utilisant une caméra calibré et les homographies correspondantes.

2.5.2 Approche de Jurie et Dhome

Une variation très importante de cette approche, est celle proposée par F.Jurie et M.Dhome ([53], [62]). A la différence de l'approche précédente, ils considèrent qu'il existe une matrice \mathbf{A} , appelé *matrice d'interaction*, telle que :

$$\mathbf{I}(t + \Delta t) = \mathbf{I}(t) + \mathbf{A}(t + \Delta t) \Delta t \mathbf{i} \quad (2.27)$$

L'équation (2.27), peut être vue comme la substitution de :

$$\mathbf{A} = (\mathbf{M}' \mathbf{M})^{-1} \mathbf{M}' \quad (2.28)$$

dans (2.26), sauf que dans cette équation, \mathbf{A} ne représente plus la matrice *Jacobienne*, et donc elle ne vient plus du calcul direct des variations d'intensité en chaque position N dans le motif de référence par rapport au modèle de mouvement choisi. Pour évaluer la matrice \mathbf{A} , il faut considérer l'équation suivante :

$$\Delta \mathbf{I}(t + \Delta t) = \mathbf{A}(t + \Delta t) \Delta t \mathbf{i} \quad (2.29)$$

L'équation (2.29) peut être vue comme :

$$\begin{aligned} (a_{11}, \dots, a_{1j}, \dots, a_{1N}, \Delta I_1) \begin{pmatrix} \Delta I_1 \\ \vdots \\ \Delta I_j \\ \vdots \\ \Delta I_N \\ \Delta I_1 \end{pmatrix} &= 0 \\ \mathbf{M} \\ (a_{i1}, \dots, a_{ij}, \dots, a_{iN}, \Delta I_i) \begin{pmatrix} \Delta I_1 \\ \vdots \\ \Delta I_j \\ \vdots \\ \Delta I_N \\ \Delta I_i \end{pmatrix} &= 0 \\ \mathbf{M} \\ (a_{n1}, \dots, a_{nj}, \dots, a_{nN}, \Delta I_n) \begin{pmatrix} \Delta I_1 \\ \vdots \\ \Delta I_j \\ \vdots \\ \Delta I_N \\ \Delta I_n \end{pmatrix} &= 0 \end{aligned} \quad (2.30)$$

où les éléments a_{ij} de la matrice \mathbf{A} sont les coefficients des n hyperplans. Cette matrice d'interaction peut être estimée au moyen de la méthode des moindres carrés.

Jurie et Dhome proposent de calculer cette matrice d'interaction \mathbf{A} à partir d'un ensemble N_p , de petites perturbations $\Delta \boldsymbol{\mu}$ autour de $\boldsymbol{\mu}_0$, c'est-à-dire en exploitant le motif de référence. Le vecteur de différences d'intensité $\Delta \mathbf{i}$ associé à chaque perturbation $\Delta \boldsymbol{\mu}$ sera calculé (figure 2.16). De telle sorte que, si $N_p > N$, on a un système surdéterminé, et il est possible d'obtenir la matrice d'interaction \mathbf{A} à partir des N_p paires $(\Delta \mathbf{i}, \Delta \boldsymbol{\mu})$, en minimisant le critère suivant aux moindres carrés:

$$\sum_{k=1}^{k=N_p} (\Delta I^k - \mathbf{A} \Delta \mathbf{i}^k)^2 \quad (2.31)$$

Dans [62], ils montrent que cette approximation par hyperplans donne de meilleurs résultats que celle de l'approximation de la matrice *Jacobienne*.

Un point important à prendre en compte avec cette approche, c'est que, pour qu'elle soit efficace, il faut exprimer \mathbf{x} , coordonnées pixel des points de la cible, dans un repère lié à la cible.

Soit $\mathbf{u} = (u, v)$ les coordonnées dans le repère image, donc la transformation entre le repère cible (\mathbf{O}_c) et le repère image (\mathbf{O}_i) est donné par :

$$\mathbf{u} = (u, v) = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) \quad (2.32)$$

où le mouvement relatif entre les deux repères est défini par le vecteur des paramètres de mouvement $\boldsymbol{\mu}$. En ce moment, il faut rappeler que $\boldsymbol{\mu}$ est exprimé dans les unités du repère image.

En considérant que \mathbf{f} est inversible, on a :

$$\mathbf{x} = \mathbf{f}^{-1}(\mathbf{u}; \boldsymbol{\mu}) \quad (2.33)$$

Dans l'équation (2.33), on considère que la fonction \mathbf{f}^{-1} fait implicitement les changements de repère sur les paramètres du modèle de mouvement, quand cela est nécessaire. La figure 2.17a montre ces transformations.

Dans le processus d'apprentissage on perturbe $\boldsymbol{\mu}_0$ avec un vecteur aléatoire $\Delta\boldsymbol{\mu}$, c'est-à-dire, qu'on obtient un nouveau vecteur de paramètres de mouvement $\boldsymbol{\mu}'$, tel que :

$$\boldsymbol{\mu}' = \boldsymbol{\mu}_0 + \Delta\boldsymbol{\mu} \quad (2.34)$$

Soit \mathbf{x} un point quelconque dans le repère cible déplacé par $\boldsymbol{\mu}_0$. En exploitant (2.32), soit \mathbf{u} les coordonnées de ce point dans le repère image.

$$\mathbf{u} = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}_0) \quad (2.35)$$

Sous la perturbation $\Delta\boldsymbol{\mu}$, le point \mathbf{x} se déplace dans le repère image. Appelons \mathbf{u}' la nouvelle position dans le repère image, c'est-à-dire :

$$\mathbf{u}' = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}') \quad (2.36)$$

Soit \mathbf{x}' la position du même point \mathbf{x} avant la perturbation (c'est-à-dire dans le repère cible au $\boldsymbol{\mu}_0$) mais dans le repère cible au $\boldsymbol{\mu}'$ (figure 2.17b). D'après (2.32) on a :

$$\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{u}; \boldsymbol{\mu}') \quad (2.37)$$

Finalement, en remplaçant (2.36) dans (2.37) on a :

$$\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}_0), \boldsymbol{\mu}') \quad (2.38)$$

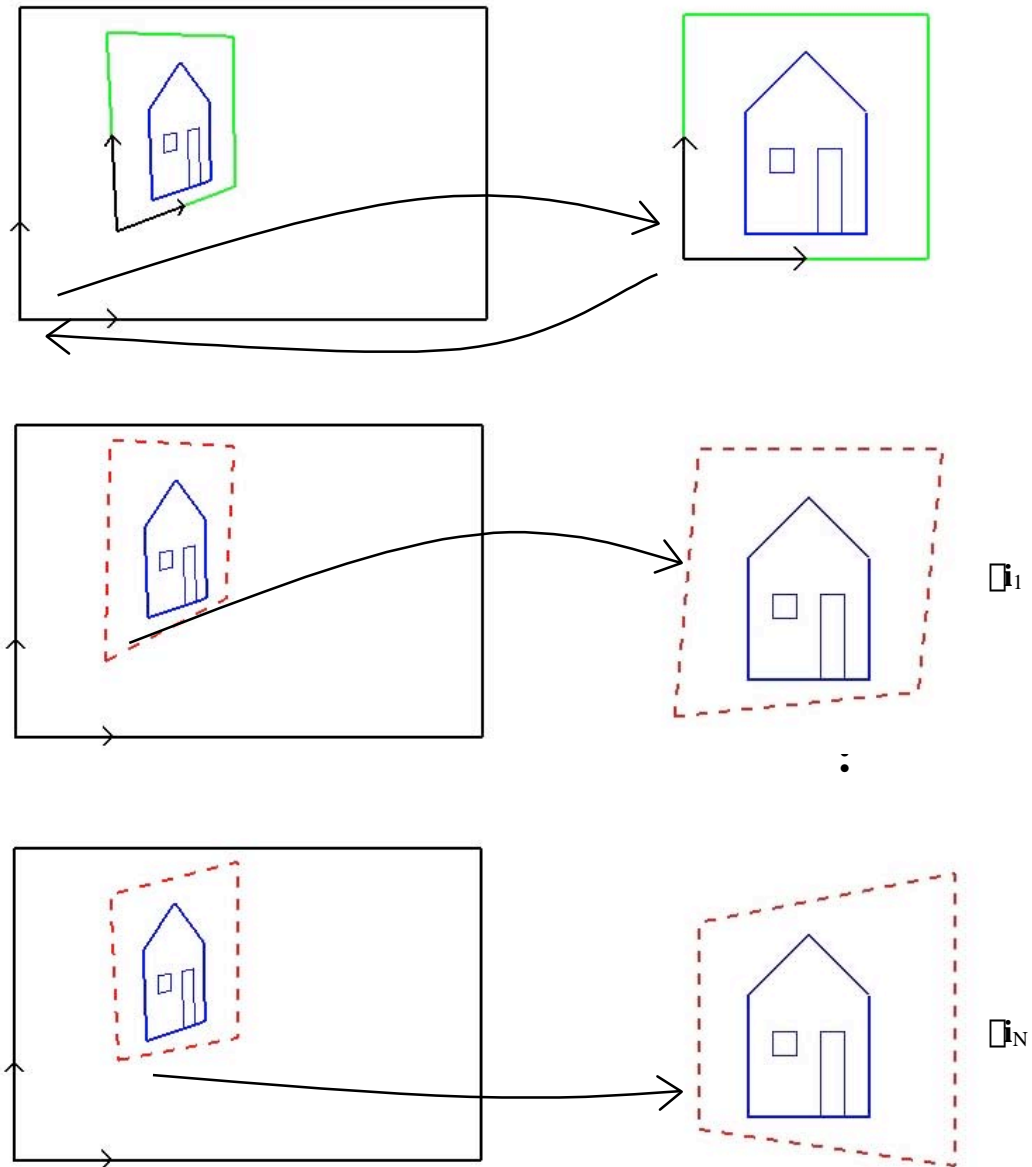


Figure 2.16. Apprentissage de A fait dans la méthode de Jurie et Dhome: définition de la région cible en haut, et variations aléatoires de la position de A_1 à A_N . A gauche, dans le repère image. A droite, dans le repère cible.

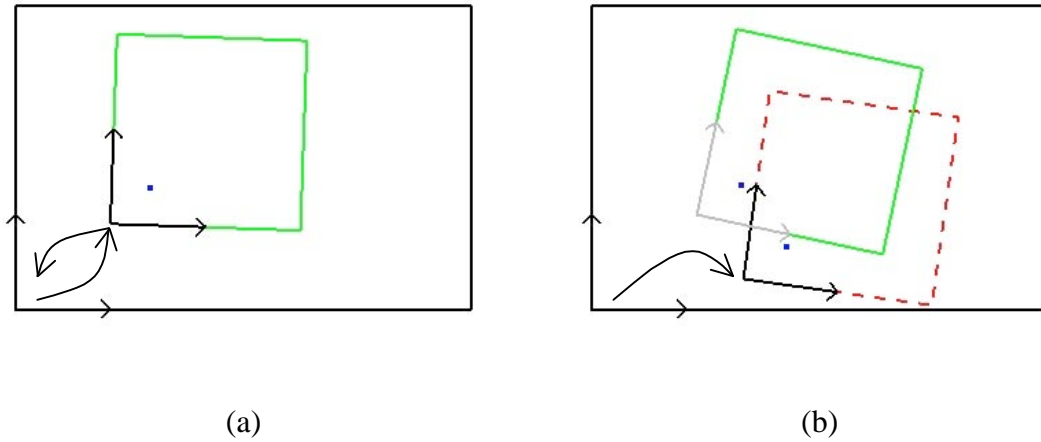


Figure 2.17. Fonctions de transformation entre le repère cible et le repère image (a) transformations entre le repère image et le repère cible au μ_0 , (b) transformation entre le repère image et le repère cible au $\mu_0 + \Delta\mu$.

Avec l'équation (2.38) on peut calculer le déplacement d'un point \mathbf{x} sous une perturbation $\Delta\mu$, dans le repère cible au μ_0 .

Tout ça veut dire que, une fois calculée l'image de différences $\Delta\mathbf{i}$, on peut estimer, avec l'équation (2.29), la perturbation $\Delta\mu$ qui a pu produire $\Delta\mathbf{i}$ dans le repère cible au μ_0 , et donc à partir de (2.38) estimer le déplacement de correction \mathbf{x}' à faire pour obtenir une approximation pour $\Delta\mathbf{i} = 0$ (figure 2.18). Cependant, le déplacement \mathbf{x}' est exprimé dans le repère cible au μ_0 . Donc, il faut passer du repère cible au μ_0 au repère cible au μ .

Soit μ' , une prédiction à un instant donné t , le suivi du motif consiste à trouver μ , tel que l'équation (2.25) soit égal à 0. Comme nous avons remarqué dans le paragraphe précédent avec $\Delta\mathbf{i}$, nous pouvons obtenir \mathbf{x}' , par rapport au repère cible avec le mouvement μ_0 ; donc pour obtenir \mathbf{u}' , c'est-à-dire la correction de \mathbf{u} , il faut substituer \mathbf{x}' par l'équation (2.38) dans (2.32).

$$\mathbf{u}' = \mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{f}(\mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu_0), \Delta\mathbf{i}), \mu_0) \quad (2.39)$$

Donc, avec cette approche il est nécessaire d'appliquer trois transformations consécutives pour obtenir la correction aux paramètres de mouvement. Cela vient principalement du fait que la matrice d'interaction (\mathbf{A}) a été apprise avec des perturbations $\Delta\mu$ autour de μ_0 , ce qui fait que les variations sont liées à cet instant t_0 . Plus précisément les paramètres ne sont pas indépendants des transformations entre les deux repères. Et c'est clair que, en calculant de cette manière deux matrices \mathbf{A} à différents instants t , elles seront différentes et donc ne seront pas comparables.

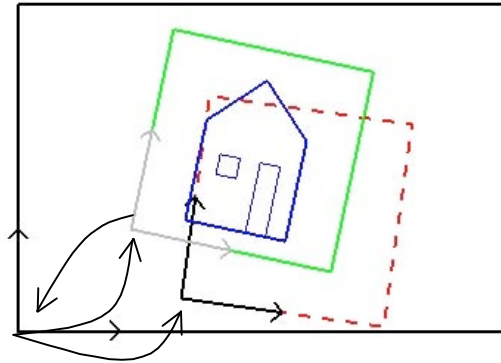


Figure 2.18. Transformation entre les repères, pour obtenir le mouvement apparente à partir de la différence.

Pour éviter de faire les transformations décrites dans le paragraphe précédent, nous proposons d'utiliser des vecteurs de variations liés directement à la cible. C'est-à-dire, utiliser des variations aléatoires $\Delta \mathbf{x}$ dans le repère cible pour l'apprentissage de la matrice d'interaction \mathbf{A} (figure 2.19). Par exemple, si la cible est un rectangle, il est possible de définir les vecteurs unitaires $(\hat{\mathbf{i}}, \hat{\mathbf{j}})$ dans le repère cible, comme les dimensions du rectangle (figure 2.19b) et donc, les variations peuvent être de 10% de la taille de ces vecteurs dans ce même repère (figure 2.19c).

Ce changement de fonctions de vecteurs des variations peut être justifié par le fait que, contrairement à l'approche de Hager et Belhumeur, dans l'approche de Jurie et Dhome le modèle de mouvement ne joue aucun rôle dans le calcul de la matrice \mathbf{A} .

De plus, le mouvement relatif de la cible par rapport à l'image courante est décrit par $\boldsymbol{\mu}$, qui n'est autre chose qu'une transformation entre le repère cible et le repère image ; donc $\boldsymbol{\mu}$ est une fonction de quelques paramètres dans le repère cible, comme par exemple les coins du rectangle englobant la cible.

Ainsi, en substituant $\Delta \boldsymbol{\mu}$ par $\Delta \mathbf{x}$ la fonction objectif à minimiser restera comme :

$$\sum_{k=1}^{k=Np} \left(\Delta \mathbf{x}^k - \mathbf{A} \Delta \mathbf{i}^k \right)^2 \quad (2.40)$$

Maintenant, le vecteur $\Delta \boldsymbol{\mu}$ sera calculé en fonction de $\Delta \mathbf{x}$ à partir de \mathbf{f} et non directement à partir de l'équation (2.30).

$$\Delta \mathbf{u} = \mathbf{f}(\mathbf{x} + \Delta \mathbf{x}; \Delta \mathbf{i}_0) \quad (2.41)$$

La principale contribution de ce changement de repère c'est l'indépendance de la matrice \mathbf{A} aux conditions initiales dans la phase de l'apprentissage, et subséquemment la récupération de corrections au vecteur $\boldsymbol{\mu}$ par une seule fonction de transformation (2.41). En plus, du fait de l'indépendance aux conditions initiales, les valeurs de différentes matrices d'interaction pourront être comparées et même, pourquoi pas, fusionnées sous certaines conditions.

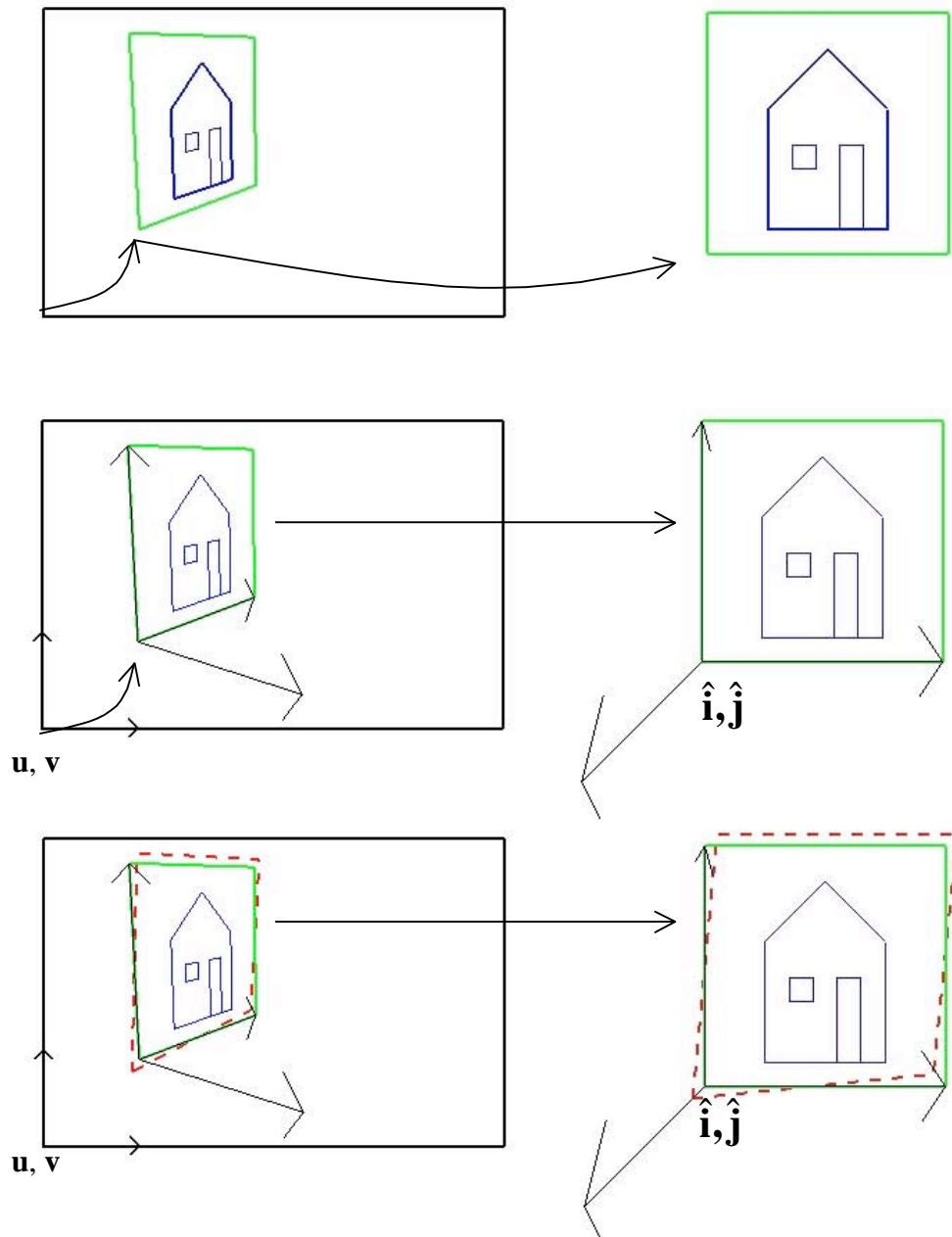


Figure 2.19. Notre méthode d'apprentissage de la matrice d'interaction dans le repère lié à la cible. A droite, dans le repère cible. A gauche, dans le repère image. En haut, la cible. Au milieu, création du repère lié à la cible. En bas, perturbations pour l'apprentissage.

2.5.3 Calcul de la Matrice d'Interaction

Nous sommes intéressé par des mouvements avec distorsions projectives, donc nous utiliserons par la suite une fonction f définie par une homographie \mathbf{H} entre les deux repères, comme il est décrit ci-dessous:

$$\mathbf{u} = f(\mathbf{x}; \boldsymbol{\mu}) \quad \mathbf{u} = \mathbf{H}\mathbf{x} \quad \mathbf{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \quad (2.42)$$

avec $\boldsymbol{\mu} = (a, b, c, d, e, f, g, h)$. Une homographie peut être calculée à partir de quatre points coplanaires, par exemple les coins d'un quadrangle.

Une façon d'obtenir les variations requises pour l'apprentissage de la matrice d'interaction \mathbf{A} , c'est de partir des variations des positions des quatre coins qui définissent la région (par exemple le quadrangle), et de cette façon, obtenir les différences entre l'homographie pour le mouvement $\boldsymbol{\mu}_0$ et pour le mouvement incluant une perturbation $\boldsymbol{\mu}_0'$.

La méthode la plus utilisée pour choisir les N positions dans le motif de référence, c'est l'échantillonnage régulier du motif. Mais il existe plusieurs autres options, par exemple dans [61] il est proposé d'échantillonner le motif en utilisant des ellipses concentriques ; il est possible aussi d'utiliser des positions aléatoires pris à l'intérieur du motif. Même, on peut faire un échantillonnage régulier ou irrégulier seulement sur les bords de l'objet à suivre.

De plus, la fonction f n'est pas seulement le niveau d'intensité de l'image, mais peut aussi être donnée par le gradient de l'intensité, ce qui fera de cette méthode, une option pour le suivi de contours. Malheureusement, avec la dernière option, la détection automatique de l'objet à suivre ne pourra pas être fait aussi facilement que comme pour les motifs texturés, en appliquant une mesure de corrélation comme celles évoquées au début de la section 2.5.

Signalons quelques points très importants à prendre en compte pour l'utilisation de cette approche :

- nous avons pu remarquer que, comme les composantes de $\nabla \mathbf{I}$ proches d'un coin seront les plus importantes à l'heure de calculer la variation de sa position, la méthode sera plus stable ; même dans certaines limites, la méthode sera possible alors de traiter avec des petites occultations ou des différences plus importantes que celles apprises lors de l'apprentissage de la matrice \mathbf{A} .
- comme il est montré dans la figure 2.20 et dans la figure 2.21, une analyse de la cible est nécessaire, afin d'éviter les zones homogènes dans la région d'intérêt ; du fait de la méthode de résolution du système (2.30), ces zones feront des points instables.
- la complexité dans l'apprentissage de \mathbf{A} , dépend directement du nombre de positions N dans le motif de référence et du nombre N_p de variations $\nabla \boldsymbol{\mu}$; malheureusement, la

stabilité de la méthode dépend de ces mêmes paramètres. Cette complexité, plus la nécessité d'analyser la cible pour éviter les zones homogènes, font que l'apprentissage en ligne reste pour nous, un défi.

Finalement, les méthodes de différences des motifs peuvent être utilisées, seulement pour des régions planes or quasi planes avec des petits mouvements, et donc pour de petites augmentations du temps Δt et de ce fait, seulement des petites variations du vecteur de mouvements $\Delta \mu$. Heureusement, une fois réalisé l'apprentissage, la méthode est très rapide, donc elle peut traiter toutes les images d'un flux vidéo standard.

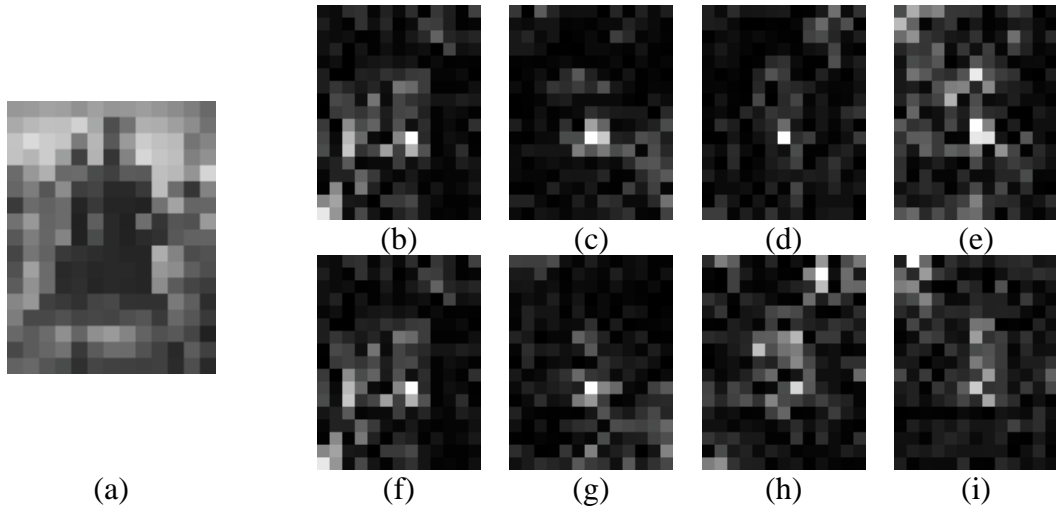


Figure 2.20. Cible et matrice d'interaction : a) région d'intérêt, de b) à i) valeurs de la matrice **A** pour les huit paramètres de mouvement, où on peut apprécier que les zones homogènes dans a) génèrent des instabilités.

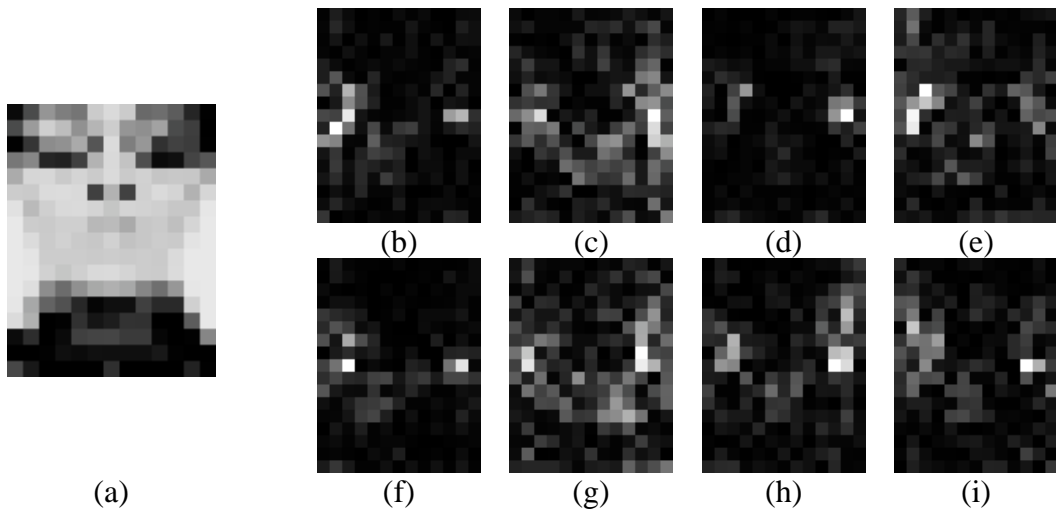


Figure 2.21. Cible et matrice d'interaction : a) région d'intérêt, de b) à i) valeurs de la matrice **A** pour les huit paramètres de mouvement.

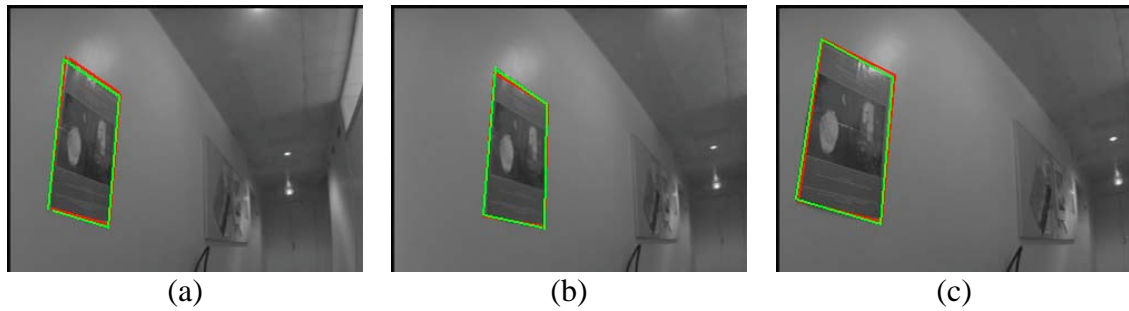


Figure 2.22. Suivi d'une affiche dans un couloir avec différences de motifs

Dans la figure 2.22, nous montrons le suivi d'une affiche dans un couloir, par la méthode de suivi par différences de motifs. Les lignes en pointillés (rouge) représentent l'estimation initiale, et les lignes continues (verte), l'ajustement de la position de la cible (vecteur de mouvement) calculée par l'équation (2.27).

2.6 Conclusions

Dans ce chapitre nous avons détaillé la formulation des quatre méthodes de suivi visuel que nous utiliserons dans nos applications.

Le suivi par des contours actifs est adapté pour des cibles de forme quelconque, éventuellement déformables ; en plus de notre proposition pour l'incorporation de forces électriques, nous avons proposé aussi deux variantes : a) la formulation de contraintes en termes d'énergie pour restreindre l'évolution du *snake* afin de respecter certaines contraintes de forme, ainsi que b) une approche pour le suivi de régions avec contours actifs dans des images couleur, qui consiste en la formulation de l'énergie externe afin de prendre en compte les caractéristiques statistiques d'une région segmentée auparavant.

Le suivi par corrélation 1D est une extension d'une méthode de reconnaissance de lignes dans des images fixes proposée précédemment dans le groupe. Nous l'avons adaptée pour en faire une méthode de suivi rapide d'un ensemble de lignes ou de courbes.

La méthode de suivi d'un ensemble des points par la distance de Hausdorff a été développée précédemment dans le groupe ; nous l'avons seulement intégrée dans notre module de suivi.

Finalement, le suivi par différences de motifs est adapté des travaux de Jurie et Dhome. Nous proposons le changement de repère pour la matrice d'apprentissage, à un repère lié à la cible afin d'éviter les multiples transformations de repère qui doivent être faits dans la méthode initiale, et ainsi afin de donner la possibilité de combiner plusieurs matrices d'interaction.

Chapitre 3

Caractérisation de Méthodes de Suivi

Dans ce chapitre nous allons analyser les caractéristiques de chacune des méthodes de suivi visuel décrites dans le chapitre précédent. Nous ferons aussi une comparaison qualitative de ces principales caractéristiques, pour ensuite déterminer les contextes où chaque méthode peut être utilisée. Par la suite nous caractériserons les cibles et environnements par une mesure de sa complexité, afin de trouver les meilleures méthodes de suivi pour chaque contexte. Finalement nous analyserons la robustesse des méthodes face à des rotations, translations et changements d'échelle de la cible suivie dans la séquence d'images.

3.1 Caractérisation des méthodes de suivi

Afin de caractériser les méthodes de suivi visuel et leurs limitations, il est possible d'analyser les différentes techniques à partir des quatre éléments suivantes [113] :

- la représentation de la cible,
- la représentation de l'observation,
- la génération des hypothèses et
- la mesure des hypothèses.

La représentation de la cible sert essentiellement à différencier la cible des autres objets. Elle peut inclure différentes modalités, comme : la forme, la géométrie, le mouvement et l'apparence. Ces modalités caractérisent la cible dans l'espace d'état, soit implicitement, soit explicitement. Cependant, trouver une bonne représentation pour les entités est un des problèmes fondamentaux de la vision par ordinateur. Généralement, des représentations concises sont adoptées pour faciliter l'efficacité. Par exemple, la

représentation d'une cible peut exploiter seulement des formes paramétriques ou des distributions des couleurs. Afin d'avoir une description plus discriminante de la cible, il est néanmoins souvent nécessaire d'exploiter deux ou plus de ces caractéristiques.

Le mouvement peut aussi faire partie de la représentation, qui peut ainsi décrire l'évolution d'un processus dynamique. La dynamique de la cible peut être incorporée dans sa représentation. Cela permettra par exemple, de différencier différents objets par leur mouvement ; si deux objets ont la même forme (par exemple, deux véhicules), sans connaissance a priori sur la dynamique de leurs mouvements, il sera très difficile de suivre correctement chacun d'eux, quand ils sont très proches dans l'espace d'état.

La représentation de l'observation est très proche de la représentation de la cible. Elle est définie par les caractéristiques extraites de l'image en vue de les comparer avec la représentation de la cible. Cette comparaison doit permettre de statuer sur la présence de la cible dans l'image. Donc, si la cible est caractérisée par sa couleur, certains modèles de distribution de la couleur pourront être utilisés comme espace d'observations de la cible.

L'opérateur qui extrait la représentation de l'observation peut être appliqué dans une ou plusieurs régions d'intérêt spécifiques ; la vitesse d'une méthode dépendra directement de la complexité de la représentation de la cible, mais aussi de la complexité des régions d'intérêt sur lesquelles l'observation est extraite (uniforme ou non, etc.).

La génération des hypothèses est la production d'états hypothétiques associant cibles et observations, états fondés sur les précédentes estimations de la représentation de la cible et sur la représentation courante de l'observation. Pour toute méthode de suivi visuel, pour chaque image de la séquence, la génération des hypothèses peut être divisée en deux parties :

- (1) une hypothèse initiale est générée sur la position de la cible dans l'image courante ; cette hypothèse est donnée par prédiction, en exploitant la position dans la ou les images précédentes, et un modèle de dynamique. L'estimation du mouvement de la cible est générée par un filtre de Kalman ou par une autre méthode d'estimation.
- (2) ensuite, la génération et l'évaluation des hypothèses permettront d'ajuster cette hypothèse initiale aux données de l'observation courante.

La mesure de l'hypothèse permet d'évaluer la correspondance entre l'état courant de la cible et son observation. Mesurer une hypothèse peut être un défi quand cette mesure se fonde sur des paramètres de forme, et que la cible évolue dans un fond encombré.

Cette mesure peut être exploitée lors de deux étapes du processus. (1) Dans une phase de *prédiction*, on veut mesurer avec quelle probabilité un état hypothétique de la cible pourra générer telle observation. (2) De la même façon, dans la phase de *vérification*, il s'agit de mesurer, étant donné une certaine observation image, quelle hypothèse sera plus vraisemblable pour produire telle observation image.

Généralement, les modèles probabilistes et l'inférence bayésienne sont exploités pour associer prédiction et vérification. Néanmoins, souvent, on ne peut pas obtenir des mesures de probabilité exactes ; on peut seulement les approximer par des pseudo probabilités. Par exemple, le suivi de motifs prend souvent la SSD (somme de

différences au carré) comme mesure. Plus petite est la mesure de SSD, plus haute sera la probabilité de l'hypothèse.

Dans la suite nous allons comparer en fonction des caractéristiques définies ci-dessus, les méthodes de suivi visuel, décrites dans le chapitre précédent, méthodes fondées soit sur le contour de la cible, soit sur la région correspondante dans l'image.

Il est important de remarquer, que pour toutes les méthodes, dans la génération des hypothèses, il est toujours possible de faire une estimation du mouvement, à partir de l'historique des mouvements précédents, à condition que les calculs soient suffisamment rapides pour ne pas affecter ni les mesures des hypothèses, ni la réponse en temps réel requis généralement pour un système de suivi visuel.

Les méthodes que nous avons présentées, ne permettent pas facilement de conserver plusieurs hypothèses sur la position de la cible dans l'image courante, comme le fait les méthodes s'inspirant de la condensation. Nos approches ne sont donc pas multi-hypothèses.

3.1.1 Contours Actifs

La méthode de suivi par snakes ou contours actifs, est une méthode de suivi à partir des contours. Même s'il existe des variations de cette méthode, qui incorporent l'information de la région à suivre, cela reste une méthode qui cherchera les cibles à partir de leurs contours.

- *Représentation de la cible*

Une cible est définie par son contour, mais, la plupart des systèmes fondés sur les *snakes* n'utilisent pas une représentation explicite de la forme que doit avoir ce contour : au contraire on laisse évoluer le contour actif jusqu'à ce qu'il converge.

L'état initial de la cible est défini sur la première image de la séquence, une fois que la minimisation de l'énergie totale du *snake* est atteinte. Le suivi consistera ensuite, à trouver pour chaque instant t de la séquence d'images, la position du snake qui minimise son énergie, fonction d'un potentiel externe.

Dans une variante de la méthode, d'autres paramètres sont introduits dans la représentation de la cible, afin de restreindre l'évolution du contour ; citons les contraintes de forme que nous avons d'écrites dans le chapitre précédent (figure 2.6).

Dans certaines applications, il est possible aussi de définir la cible à partir d'autres caractéristiques que leur forme ou contour, par exemple, en utilisant la couleur ou la texture de la région définie par le contour (figure 2.9). Cependant la méthode de suivi par des *snakes* n'utilisera pas seulement cette information pour trouver les limites de la région, ce qui la rendra plus robuste aux situations dans lesquelles la cible n'a pas une couleur ou une texture uniforme (par exemple du fait d'ombres ou de reflets).

Comme nous le verrons dans l'analyse des performances, les variantes évoquées ci-dessus sur la représentation de la cible par un *snake* (contraintes de forme, couleur de la région englobée) rendent plus difficile, les phases de génération et évaluation des hypothèse, puisqu'elles compliquent les calculs et augmentent le temps requis pour

chaque itération. De ce fait, il pourrait s'avérer plus difficile de faire un suivi en temps réel.

- *Représentation de l'observation.*

C'est dans la fonction du potentiel externe, qu'est définie la représentation de l'observation. En effet, cette fonction définit les caractéristiques qui vont créer un potentiel attractif sur la cible. Par exemple, dans la plupart des cas, c'est le gradient de l'intensité lumineuse dans l'image qui suffit pour représenter l'observation (figure 2.7e). Cependant, d'autres fonctions de potentiel peuvent être définies, par exemple, des gradients dans des espaces couleur (figure 2.7f et g), ou des gradients spécifiques sur certaines caractéristiques (figure 2.7h), comme nous l'avons proposé pour la localisation et le suivi de la route, dans la section 2.2.3.

- *Génération des hypothèses.*

Tout d'abord, l'hypothèse d'initialisation dans l'image courante, provient du résultat de la minimisation de l'énergie dans l'image précédente.

Ensuite, la génération des hypothèses sur l'évolution du contour de la cible, dépend de la méthode choisie pour la résolution du système dynamique. S'il est résolu par des méthodes variationnelles, la solution des équations de mouvement produira, les hypothèses ; plus précisément, elles seront définies par l'intervalle de variation choisi pour les paramètres de la représentation de la cible. Par contre, s'il est résolu à partir d'une méthode non directe de minimisation de l'énergie, telle que la programmation dynamique [4], ce sera la définition des zones de recherche qui produiront ces hypothèses. Par exemple, si on définit, la zone de recherche du *snake*, comme des lignes droites perpendiculaires à son état courant (figure 2.2), ces hypothèses peuvent être les positions de points de contrôle à une certaine distance \square , à l'intérieur ou à l'extérieur du contour sur ces lignes droites.

- *Mesure des hypothèses.*

Dans les contours actifs, la mesure de l'adéquation d'une hypothèse à l'observation courante, est fournie par la solution de l'équation de mouvement ou par le calcul de l'énergie, soit par les méthodes variationnelles, soit par les méthodes non directes de minimisation. Par exemple, en résolvant le système d'équations de mouvement, tant que les variations entre deux hypothèses (contours aux itérations $t-1$ et t) restent plus grandes qu'un seuil donné ou tant que le temps maximal n'est pas encore dépassé, on peut continuer à produire des hypothèses (variations) pour ajuster la solution. Dans le cas de la programmation dynamique, tant qu'il existe une configuration qui permette de diminuer l'énergie, on peut continuer à produire des hypothèses, jusqu'à atteindre un nombre maximal d'itérations, un seuil défini sur le temps d'exécution ou une diminution de l'énergie inférieure à un seuil.

3.1.2 Corrélation 1D

Le suivi avec corrélation 1D, est également une méthode fondée sur les contours. Même si la cible peut être un ensemble de courbes délimitant une région (par exemple, un quadrangle ou une forme plus complexe), la méthode suivra cette cible à partir de ces contours.

- *Représentation de la cible*

La cible est définie par une seule ou un ensemble de courbes : par exemple, une simple ligne droite pour suivre les transitions mur / sol ou mur / plafond, un ensemble de lignes droites qui forment un quadrangle, pour suivre une porte, un panneau de signalisation ou une affiche murale, ou une structure plus complexe [39] faite d'un ensemble de courbes et lignes droites.

- *Représentation de l'observation.*

A la différence des contours actifs, les hypothèses seront générées à partir d'une mesure de corrélation, entre un modèle de ligne et l'observation courante. Ce modèle habituellement utilise le gradient de l'intensité lumineuse dans l'image, obtenu par exemple, à partir d'un détecteur de contours (Sobel, Canny, Deriche...) ; mais il peut être utilisé n'importe quelle autre mesure de corrélation pour représenter l'observation.

- *Génération des hypothèses.*

La génération des hypothèses pour ajuster l'hypothèse initiale, peut exploiter diverses méthodes, mais pour la plupart d'entre elles, c'est à partir des lignes droites perpendiculaires aux courbes qui représentent la cible, qu'on va chercher dans l'observation, des zones maximales de corrélation, qui dans la plupart des approches, sont des fortes variations de l'intensité ou du gradient d'intensité. A partir de l'ensemble de ces points de forte corrélation, de nouvelles hypothèses sont générées aléatoirement par la méthode de RANSAC. Même si plusieurs hypothèses sont habituellement générées, seule la première qui satisfait le test d'acceptation, sera la solution.

- *Mesure des hypothèses,*

La mesure de la pertinence de l'hypothèse va exploiter la définition de la configuration de la cible. Les hypothèses générées seront évaluées : celles qui ne satisfont pas la configuration donnée de la cible, seront rejetées. Par exemple dans notre implémentation, c'est par la méthode de RANSAC, que seront générées aléatoirement, mesurées et rejetées les différentes hypothèses.

3.1.3 Ensemble de points.

La méthode de suivi d'un ensemble de points par la distance de Hausdorff est une méthode de suivi basée sur les régions, les points étant extraits dans une région spécifique de l'image.

- *Représentation de la cible*

La cible est définie par un ensemble de points caractéristiques à l'intérieur ou au bord d'une région, définie dans notre implémentation, par un rectangle englobant la cible. Il peut s'agir de points de contour, de points d'intérêt, ou autres points caractéristiques (couleur, Harris, etc.) extraits par des détecteurs spécialisés, doués de propriétés d'invariance aux transformations projectives (voir section 2.4).

- *Représentation de l'observation.*

C'est aussi, à partir des détecteurs de points d'intérêt utilisés dans la définition de la cible, que sera défini l'espace d'observations.

- *Génération des hypothèses.*

La génération des hypothèses est faite à partir de zones de recherche, définies autour de l'hypothèse initiale en fonction du mouvement de la cible, si ce mouvement est inclus dans leur représentation. Par exemple, pour des cibles qui bougent plutôt horizontalement, il est possible d'incorporer cette information dans la génération des hypothèses. Mais, dans le pire de cas, quand on ne connaît pas à priori le mouvement possible de la cible, une génération des hypothèses à partir d'une zone de recherche en spirale, autour de l'hypothèse initiale, donne des résultats satisfaisants.

- *Mesure des hypothèses.*

Les hypothèses seront mesurées, par la distance de Hausdorff, entre l'ensemble de points qui représentent la cible et ceux observés dans la zone de l'image qui caractérise l'hypothèse examinée.

3.1.4 Différences de Modèles

La méthode de différences de motifs est aussi une méthode de suivi basée sur les régions.

- *Représentation de la cible.*

La cible est définie par une région d'intérêt, correspondant à une zone plane ou quasi-plane dans la scène 3D. Cette région sera sous-échantillonnée sous certaines conditions, ce qui donnera un motif, en principe texturé, comme ceux montrés dans la figure 2.20a et dans la figure 2.21a.

- *Représentation de l'observation.*

Les observations seront définies à partir de l'hypothèse courante, comme cela a été décrit dans la section 2.5. Il s'agit d'une région image, sous-échantillonnée dans les mêmes conditions que lors de la définition initiale de la cible.

- *Génération des hypothèses.*

Contrairement aux autres méthodes, pour le suivi par différence de motifs, on peut considérer qu'il n'existe qu'une seule hypothèse à chaque instant t , l'hypothèse initiale issue de l'image traitée à l'instant $t-1$, recalée grâce à un vecteur de mouvement.

- *Mesure des hypothèses.*

Pour cette méthode, *mesure* est un terme impropre, puisqu'il n'existe pas à proprement parler d'évaluation de l'hypothèse sur la position de la cible dans l'image. La mesure effectuée sur l'hypothèse courante consiste à calculer un vecteur de mouvement $\Delta\mu$ de la cible à partir de la différence Δi entre représentations de la cible (le motif, comme celui présenté en figure 2.15c) et de l'observation. La magnitude du vecteur $\Delta\mu$, est le résultat de la multiplication de l'observation Δi avec la matrice d'interaction \mathbf{A} (équation (2.27)). S'il est égal à zéro, ça veut dire que l'hypothèse est correcte ; sinon, une nouvelle hypothèse sera générée par l'addition de ce vecteur $\Delta\mu$, au vecteur du mouvement de la cible μ au temps t précédent (recalage sur la position de la cible). A priori, il n'est pas nécessaire de mesurer cette nouvelle hypothèse, parce qu'elle a été générée à partir d'un apprentissage de la cible qui tient compte des conditions possibles de mouvements et de déformations de la cible. Donc le mouvement nécessaire pour rendre nul le vecteur $\Delta\mu$, est compris dans ce calcul.

Il est toujours possible d'affiner la nouvelle hypothèse générée par l'addition du vecteur $\square\mu$, en faisant une itération sur les approximations du vecteur de mouvement μ ; cependant, les résultats obtenus sont presque imperceptibles, et difficilement différenciables des erreurs dues au bruit ou à la matrice d'apprentissage.

3.1.5 Initialisation des méthodes de suivi.

Une caractéristique qui n'est pas toujours prise en compte, quand on fait l'analyse d'une méthode de suivi visuel, est la phase d'initialisation. C'est pourtant est une phase très importante pour la comparaison de différentes méthodes. Cette caractéristique est très liée d'une part à la représentation de la cible exploitée par le module de suivi, d'autre part à la méthode mise en œuvre pour détecter la cible avant de lancer son suivi. Détection et suivi sont deux processus liés, mais profondément différents dans leurs objectifs :

- La **détection** fait une analyse **globale** de l'image pour y découvrir la cible, sans aucune prédiction a priori ; c'est souvent un processus plus lent, capable de générer une description plus complète de la cible.
- Le **suivi** exploite des régions d'intérêt dans chaque image d'une séquence, régions définies par la prédiction. Donc, c'est une analyse **locale**, s'appuyant sur une description minimale de la cible. Ce faisant, le suivi peut dériver (confusion entre cible et fond) ou même perdre la cible du fait d'une mauvaise prédiction, d'une description trop sommaire de la cible ou de l'absence de vérification globale sur l'image.

Pour certaines applications, pour éviter ce manque de vérification et rendre de ce fait, plus robuste la fonction, détection et suivi sont réalisés dans un même processus. Par exemple, dans les travaux de R.Labayrade [70] sur la détection d'obstacles depuis un véhicule en milieu routier, la détection est réalisée sur chaque image à 25Hz. Dans un tel cas, si cela est nécessaire pour l'estimation de la dynamique des cibles ou pour la fusion multisensorielle, le suivi est une simple association des cibles détectées d'une image à l'autre. Une telle réactivité de la détection se paye généralement par la pauvreté de la représentation, par des simplifications et au final, par la robustesse. Néanmoins, l'évolution de la technologie (processeurs plus rapides, architectures dédiées intégrées sur le capteur) rendra sans doute possible dans un futur proche, des traitements complexes sur des images de haute résolution, à des fréquence qualifiée aujourd'hui de temps réel.

Pour notre part, nous maintenons la différence entre fonctions de détection et de suivi ; dans le Chapitre 5 , nous évoquerons plusieurs contextes applicatifs, dans lesquelles seront intégrées nos méthodes de suivi visuel. Nous allons analyser les différentes initialisations requises pour chacune des méthodes évaluées dans ce chapitre.

- Contours actifs.

Le processus d'initialisation en ce cas, dépend de la forme de la cible et du type de fond. Dans la plupart des cas, une fois détecté l'objet à suivre, une initialisation par des points de contrôle choisis sur le contour de la position détectée de la cible suffit, sauf dynamique initiale très importante. Cependant, si le fond est encombré, il est très facile d'arriver à des minima locaux de la fonction potentielle.

- Corrélation 1D.

La plupart des approches qui utilisent cette méthode, font une projection des courbes ou arêtes 3D d'une entité de la scène, afin d'obtenir la représentation initiale de la cible. Une telle projection nécessite un calibrage exact (ou fort, en opposition à faible) des paramètres de la caméra, un modèle de l'environnement 3D ou d'objets 3D qui s'y trouvent, et un module visuel de reconnaissance et localisation des entités 3D qui localiseront les cibles à suivre.

Dans [40] une méthode d'images de synthèse et CAD a été utilisée pour estimer les courbes visibles de la cible depuis une position donnée de la caméra.

Les connaissances nécessaires peuvent être plus qualitatives; une méthode dédiée détecte directement dans l'image, les courbes qu'il faudra suivre ; en ce cas, il n'est pas utile que la caméra soit calibrée. Par exemple, un module d'analyse de scènes, exploitant la recherche des points de fuite, peut trouver directement dans l'image les segments correspondants aux arêtes mur/sol ou mur/plafond, si le robot navigue dans un couloir. Dans [64] les marquages sur un terrain de football sont trouvés dans l'espace de Hough, et c'est à partir de cette initialisation de lignes droites qu'un processus d'autocalibration, de construction d'une mosaïque et de suivi des lignes est initié.

Signalons ici l'intérêt des méthodes qui ne nécessitent pas de calibrage, ou qui réalisent un auto-calibrage ; elles favorisent l'exploitation de zoom, dont le calibrage fort peut se révéler au mieux fastidieux, au pire difficile.

- Ensemble de points.

Pour cette méthode, il n'est pas nécessaire de connaître a priori une définition précise de la cible. Elle peut être fournie dans une phase d'initialisation, par l'extraction d'une zone d'intérêt, correspondant par exemple à un obstacle dans une scène routière, ou à un intrus mobile dans une scène de vidéo-surveillance. La représentation initiale de la cible est alors donnée par les points d'intérêt extraits de cette zone. Cependant, une connaissance des caractéristiques de mouvement de la cible, peut être exploitée afin de réduire l'espace de générations des hypothèses (direction privilégiée de mouvement).

- Différences des motifs.

A la différence de la méthode précédente, pour cette méthode, il est souhaitable de connaître un modèle de la cible : apparence dans un point de vue standard, et type de déformations à laquelle elle peut être soumise. Comme le processus d'apprentissage peut être long, il peut ainsi être réalisé hors ligne. Le processus d'initialisation nécessitera alors de reconnaître la région d'intérêt dans l'image courante, correspondant à la cible ; cela peut être fait par différentes méthodes de reconnaissance comme par exemple celle utilisée dans [51].

Malgré la nécessité de faire un apprentissage, il est possible d'extraire un motif intéressant en ligne, comme dans les travaux de CLADY, mais cela implique d'optimiser la phase d'apprentissage (300ms pour CLADY), ou de simplifier le modèle de mouvement de la cible (uniquement translation et changement d'échelle dans [28], dans les cas où il est nécessaire de suivre cet objet tout de suite.

3.2 Détection et reprise d'erreurs

Plus qu'une caractéristique, la détection d'erreurs est une mesure importante de la performance d'une méthode de suivi visuel, ainsi qu'une manière de définir les stratégies possibles pour reprendre la fonction en cas de perte de suivi. C'est donc très important d'analyser les causes possibles d'erreurs, et de dresser la liste des erreurs qu'il sera possible de détecter pour chacune des méthodes.

3.2.1 Contours Actifs

Comme il a été analysé dans la section 3.1.1, les contours actifs génèrent des hypothèses et les évaluent à partir de l'équation de mouvement, fondée sur l'évolution de potentiel externe (figure 3.1). Cependant, bien que le système soit bien posé, il existera toujours une solution à ce système, même quand le potentiel externe est nul, du fait des forces internes qui feront évoluer le contour pour atteindre le minimum d'énergie, comme il est montré dans la figure 3.2.

Par la suite, il sera très difficile de détecter une erreur à partir de la génération ou de l'évaluation des hypothèses. En plus, il est possible que le contour reste accroché à des minima locaux, ou que la minimisation ne soit pas atteinte dans le temps max prédéterminé pour que la méthode reste en temps borné.

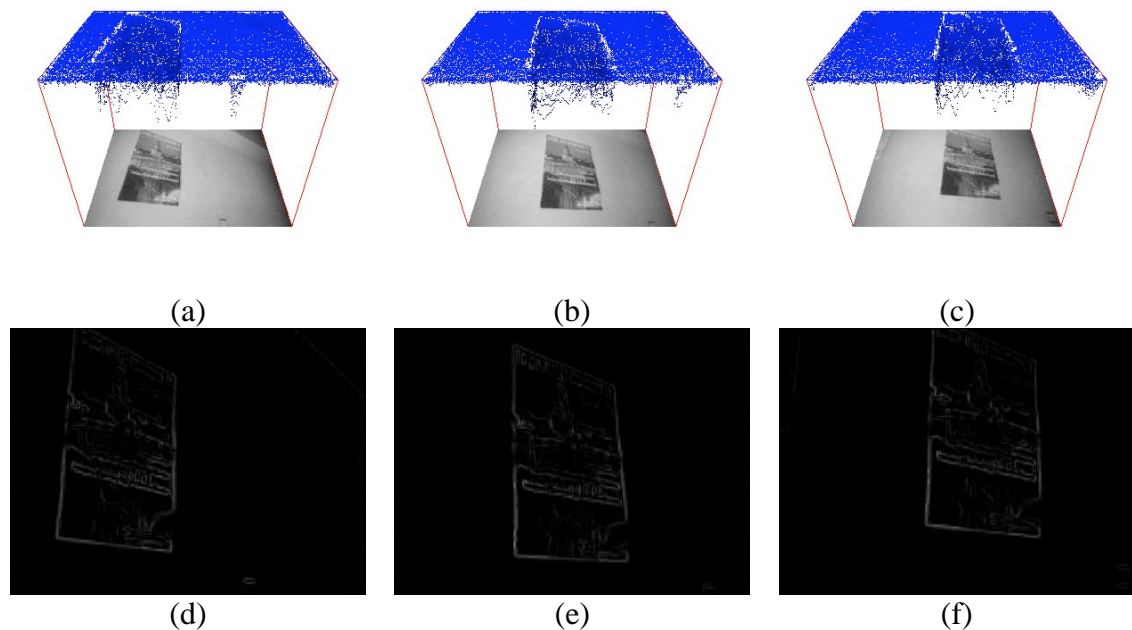


Figure 3.1. Evolution d'un potentiel externe : en haut, attracteurs créés par les images mises pour cette illustration dans un plan horizontal ; en bas images de gradients qui génèrent ces attracteurs.

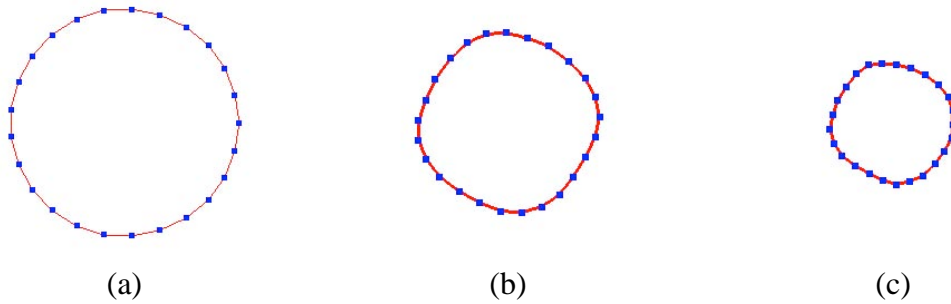


Figure 3.2. Evolution d'un contour actif dans un potentiel externe nul : a) Initialisation b) et c) résultats après quelques itérations.

Pour détecter une erreur dans l'évolution d'un contour actif, une mesure externe doit donc être incorporée. Cela peut être une mesure de corrélation dans l'espace des observations, entre la forme prédéfinie de la cible I et les hypothèses générées à l'instant donné.

Malgré cela, cette mesure ne suffit pas pour assurer un bon suivi de la cible en question, comme le montre la figure 3.3 sur le suivi d'un quadrangle.

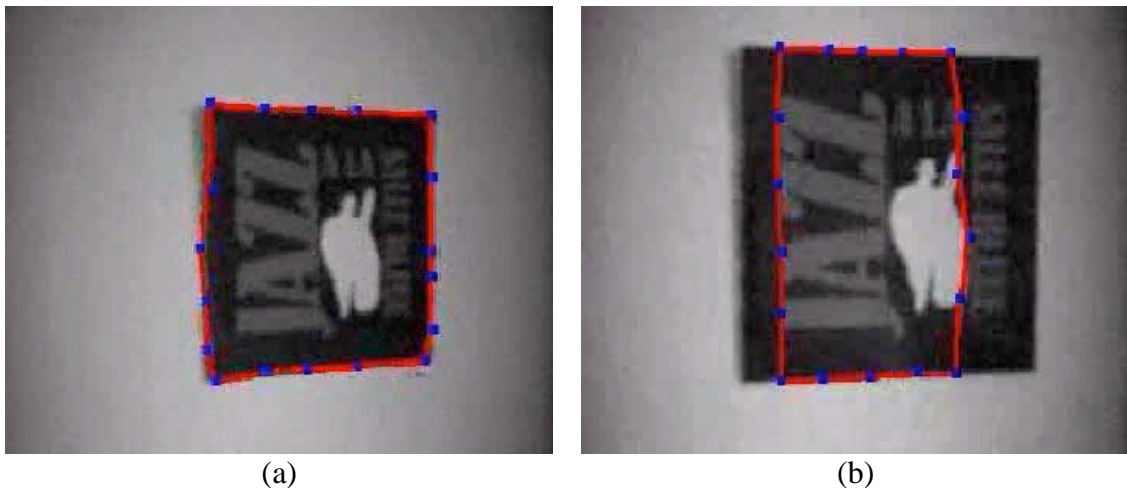


Figure 3.3. Suivi d'une affiche avec un contour actif a) convergence du contour actif sur le quadrangle et b) erreur de suivi, mais qui satisfait les conditions de forme et de gradient.

D'autre part, quand la cible n'a pas été définie comme une forme paramétrique, comme par exemple, dans le cas de la route de la figure 2.9, cette mesure n'aura pas de sens. Une manière possible de détecter des erreurs dans tous les cas, est de détecter des variations brusques de forme. Par exemple, en tenant en compte que la vitesse de la méthode est suffisamment rapide pour adapter le *snake* sur des petits changements d'intensités lumineuses entre deux images successives, une variation brusque de forme peut être vue comme une erreur de la méthode de suivi. Ces mesures peuvent être incorporées par exemple, par moyenne des moments de la surface englobée par le *snake*.

Comme la méthode ne contient pas, par elle-même, de moyens pour détecter une erreur dans la phase de suivi, la reprise de la fonction une fois qu'une erreur a été détectée, devient extrêmement compliquée. En effet, la détection survient généralement trop tard et l'état courant du *snake* n'est plus représentatif de l'état de la cible.

Pour des modèles paramétriques, il est possible de générer plusieurs hypothèses qui répondent à certains critères, comme dans le cas de la Condensation [53]. Cependant

à mesure que la complexité du modèle augmente, l'évaluation de toutes les hypothèses devient un défi. C'est pour cela que nous n'avons pas mis en œuvre une méthode multi hypothèses pour évaluer plusieurs hypothèses et finalement garder uniquement celles qui ne sont pas en erreur.

La plupart du temps il sera nécessaire de faire une relocalisation de la cible et de réinitialiser l'algorithme avec ces nouveaux paramètres. Nous montrerons un exemple de ces cas dans le chapitre 5 .

3.2.2 Corrélation 1D

Comme il a été décrit dans la section 3.1.2, la génération des hypothèses pour les méthodes de corrélation 1D, peut être faite à partir des points de corrélation maximale par des algorithmes probabilistes, ce qui rend la routine de détection des erreurs plus facile. Par exemple, nous utilisons la méthode de RANSAC (décrite au chapitre 2), pour générer et évaluer les hypothèses. Parmi les paramètres internes de l'algorithme, un seuil permet de définir le pourcentage d'erreur à l'heure de l'évaluation de chacune des hypothèses générées. En fait, deux seuils internes, très liés entre eux, sont illustrés en figure 3.4: le premier définit un nombre minimum de points corrélés, et le deuxième définit la distance maximale entre la courbe d'approximation, et les points corrélés pour considérer une bonne évaluation de l'hypothèse courante. Quand ces seuils ont été dépassés, on peut considérer que la fonction de suivi a échoué.

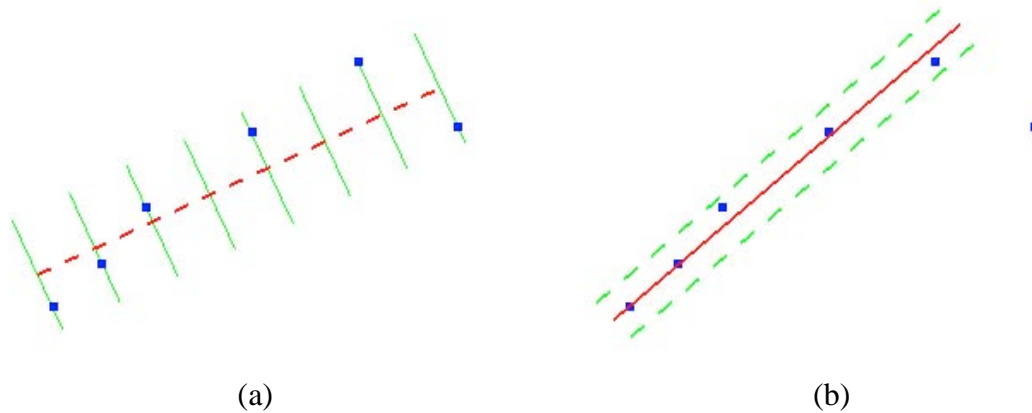


Figure 3.4. Seuils utilisés pour la détection d'erreur avec la corrélation 1D, a) nombre total de points corrélés, b) distance maximale des points à l'hypothèse courante sur l'état de la cible.

Un autre moyen, mais moins fiable pour détecter une erreur de suivi, peut être comme dans le cas de contours actifs, une variation brusque, par exemple la réduction de moitié ou plus de la longueur d'une ligne droite entre deux images successives.

Afin de reprendre l'activité une fois qu'une erreur est détectée, on peut soit augmenter le nombre de points échantillonnés le long de la courbe, soit augmenter les distances maximales de recherche le long des lignes perpendiculaires à la courbe. Cependant, dans ce dernier cas, dans le cas d'environnements encombrés, il est très facile de tomber sur des points de corrélation maximale dans le voisinage.

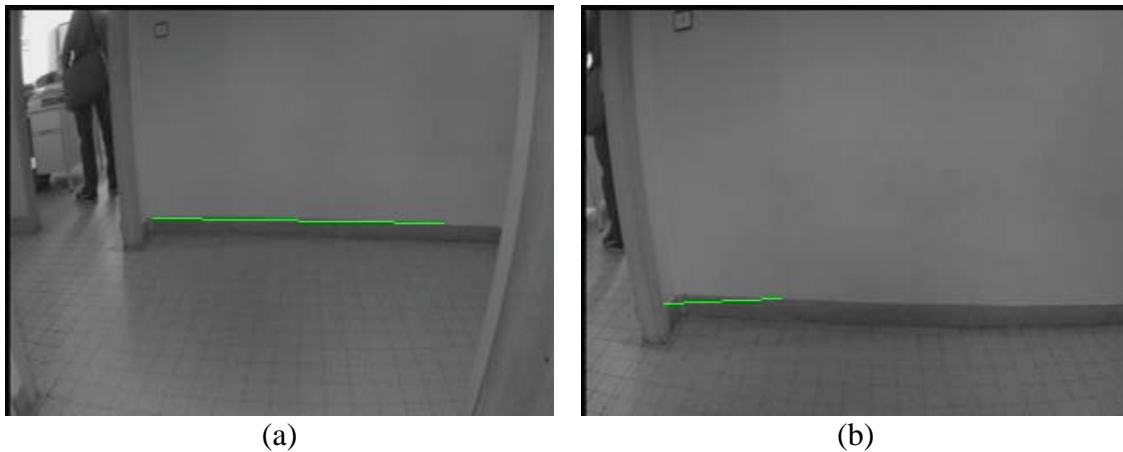


Figure 3.5. Détection d'erreur par changement brusque de la taille de la ligne entre deux images.

3.2.3 Ensemble de points

Comme nous l'avons décrit, dans la section 2.4, cette méthode de suivi, de manière similaire au suivi par corrélation 1D, contient comme paramètre un seuil interne pour déterminer et évaluer les hypothèses générées. Ce paramètre est le nombre de points appariés entre les ensembles de points extraits sur la région qui définit la cible, et sur la région correspondant à l'hypothèse courante. Ce seuil est une distance de Hausdorff partielle, qui permet de déterminer la mesure d'évaluation de l'hypothèse courante.

Dans le cas du suivi d'un ensemble de points, il est plus difficile de déterminer une variation de forme.

Quand une erreur a été détectée dans cette méthode, la méthode augmentera automatiquement la zone de recherche, pour essayer de relocaliser la cible en générant plus d'hypothèses. Cependant, comme c'est une méthode lente, au moins en comparaison des autres méthodes décrites dans ces travaux, on risque de perdre beaucoup de temps à chercher la cible. C'est pourquoi des seuils ont été incorporés pour limiter la taille maximale de la zone de recherche et le temps maximal de recherche.



Figure 3.6. Points corrélés dans la région de la cible.

3.2.4 Différences des Modèles

Même si elle ne recèle pas de seuil d'appariement entre cible et observation courantes, la méthode de suivi par différences de motifs est celle qui dispose de plus de critères internes pour détecter les erreurs faites dans la phase de suivi. La première détection d'erreur est faite à la génération des hypothèses ; c'est une mesure de corrélation par un score SSD entre la cible définie (région d'intérêt) et l'hypothèse générée. En effet, comme les variations possibles de la région cible ont été généralement apprises dans une phase hors ligne, il suffit simplement de comparer le score SSD de l'hypothèse en cours, avec les scores produits dans l'apprentissage ; s'il est très élevé par rapport aux scores appris, nous considérons cette hypothèse comme une erreur de suivi. La plupart du temps, ces erreurs sont produits par des occultations temporelles de la cible (personne ou objet qui passent devant la cible, ...).

Un deuxième critère de détection est fourni par les vecteurs de mouvement produits par la génération de nouvelles hypothèses. Les vecteurs produits à l'heure de l'apprentissage de la cible, peuvent être comparés à celui produit en ligne ; si ce dernier est plus grand que les vecteurs appris, on peut considérer que le suivi a échoué, ou que le mouvement apparent de la cible dans l'image courante, sort des limites prises en compte lors de l'apprentissage de la matrice \mathbf{A} d'interaction. Malheureusement, cette mesure n'est pas très fiable, parce que des petites occultations peuvent produire des variations plus grandes que celles apprises ; ces variations anormales affectent seulement une zone spécifique de l'hypothèse ; on peut considérer qu'avec les autres paramètres, il est possible malgré cette petite occultation partielle, de continuer à suivre la cible (figure 3.7).

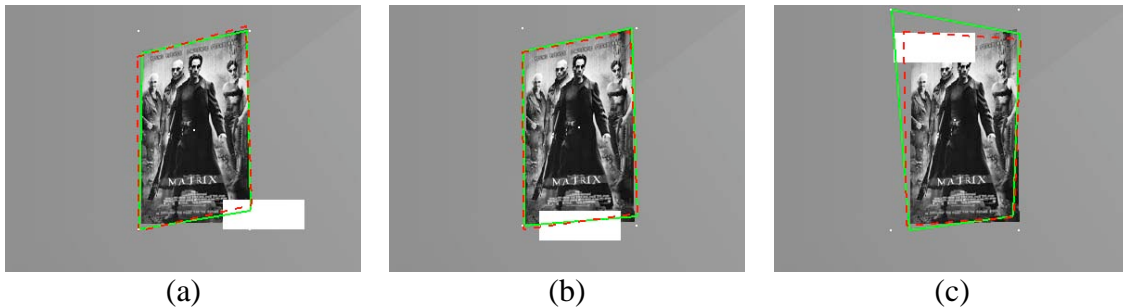


Figure 3.7. Suivi d'une cible plane avec des petites occultations

D'autres mesures externes pour la détection d'erreurs commises par la méthode de suivi par différences de motifs, peuvent être incorporées à partir de la cohérence de l'hypothèse générée à l'instant t et la région de la cible déduite de l'instant $t-1$. Par exemple, le parallélisme, la convexité, etc. Par exemple la convexité de la région de l'hypothèse peut être testé de manière très simple, à partir des quatre coins qui définissent la cible, vu que ces points sont ordonnés. Donc une mesure comme celles utilisées pour les enveloppes convexes peut être adéquate (déterminant de trois points consécutives).

La reprise de la fonction est difficile avec cette méthode. Néanmoins dans la plupart des cas, les erreurs sont dues aux occultations ; une manière de restaurer un état de la cible, consiste à reprendre sa dernière approximation valable et de relancer le suivi

à partir de cela. Si une erreur survient de nouveau, alors une relocalisation de la cible par une autre fonctionnalité est nécessaire.

3.3 Comparaison des méthodes de suivi

Nous disposons donc, de quatre méthodes de suivi visuel, qui exploitent différentes représentations des cibles, différents espaces d'observations, ainsi que différents moyens de générer et mesurer des hypothèses.

Il est assez difficile d'établir une comparaison au niveau quantitatif de ces méthodes, puisqu'elles ne s'appliquent pas aux mêmes cibles ou adopte des représentations différentes difficiles à comparer. Un autre moyen serait d'avoir une vérité terrain, par l'extraction systématique de la position réelle de la cible, sur toute une séquence ; mais cela devient extrêmement compliquée, principalement du fait que dans les situations dans lesquelles ces méthodes seront utilisées, le robot et la caméra bougent ; impossible d'utiliser des méthodes d'extraction des cibles par de simples différences d'images pour détecter les objets mobiles. La seule manière de trouver les positions réelles de la cible dans une séquence, consisterait à segmenter tous les images afin de l'en extraire. Cependant, il faudrait localiser la cible pour un grand nombre d'images dans des différentes situations, ce qui en ferait une tâche extrêmement compliquée

Par la suite, donc, nous allons comparer les méthodes d'une manière plus qualitative, l'objectif étant de caractériser, à la fin de ce chapitre, les méthodes en fonction des cibles et conditions de l'environnement pour lesquelles elles fonctionnent.

3.3.1 Fréquence.

Une des plus importantes mesures pour la comparaison des algorithmes de suivi visuel est la fréquence de travail. Cette mesure, bien sûr, dépend de l'ordinateur utilisé ; de plus, comme les progrès technologiques des ordinateurs sont de plus en plus rapides, nos mesures seront obsolètes dans très peu de temps. La comparaison relative entre les fréquences des méthodes nous permet simplement de les classer sur le critère de la vitesse moyenne sur un très grand nombre de test. La table suivante montre les résultats obtenus en moyenne pour chaque méthode ; les mesures ont été obtenues sur une station SunBlade 100 à 500Mhz.

Méthode	Fréquence
Contours Actifs	~12 - 18 Hz.
Contours Actifs avec contraintes	~10 - 16 Hz.
Corrélation 1D	~12 - 16 Hz.
Ensemble de points	~1 - 5 Hz.
Différences de motifs	~20 - 25 Hz.

Table 3.1. Evaluation de la fréquence de travail de chaque méthode de suivi.

Comme on peut voir dans la table 3.1, la méthode la plus rapide est la méthode de différences de motifs. Cette méthode a un temps de calcul fixe, qui dépend du nombre de paramètres du modèle de mouvement, et du nombre N de points à l'intérieur de la zone d'intérêt.

Ensuite, ce sont les contours actifs, qui ont la meilleure performance par rapport au temps de calcul. Ce temps, à chaque itération, dépend directement (1) du nombre de points de contrôle, (2) du nombre de positions possibles pour la génération des hypothèses (voir section 2.2), et enfin, (3) du nombre d'itérations nécessaires pour arriver au minimum d'énergie. Pour éviter, un temps de calcul très élevé dû au nombre d'itérations, on peut limiter par un seuil, le nombre maximal de celles-ci, ce qui permet de définir une borne maximale du temps de calcul. Quand les contours actifs sont contraints afin de conserver lors de leur évolution, une forme paramétrique, le temps augmente en raison de la prise-en-compte de nouveaux termes d'énergie, qui impliquent en plus de considérer ensemble certains points de contrôle, ce qui rend l'algorithme plus complexe.

Le suivi avec corrélation 1D est également une méthode très rapide de suivi. En ce cas, le temps de calcul dépend directement du nombre des points à corrélérer (donc du pas d'échantillonnage le long des courbes) et du nombre d'hypothèses pour chacun, dépendant de la distance maximale de recherche d'un point, sur les normales à la courbe suivie. Enfin, ce temps dépend aussi du nombre maximal d'hypothèses générées par l'algorithme de RANSAC. En tenant compte de ces paramètres, c'est aussi un temps borné.

Enfin, la méthode de suivi d'un ensemble de points est la plus lente. Le temps de calcul de cette méthode dépend d'une part de la taille de la fenêtre de recherche et d'autre part du nombre d'itérations faites pour localiser la cible, même s'il est possible de borner le temps de calcul, en donnant un seuil pour le nombre maximal d'itérations, cette méthode est la seule qui puisse relocaliser une cible perdue, bien sûr, sous certaines conditions.

Pour toutes ces méthodes, comme il a été mentionné dans la section 3.1.3, il est possible d'obtenir une réduction du temps de calcul, en réduisant la zone de recherche pour la génération des hypothèses par l'introduction dans la définition de la cible, du modèle de mouvement de celle-ci.

3.3.2 Cibles et environnements

La représentation des cibles, pour chaque méthode, est différente ; il est nécessaire d'analyser pour quelles sortes de cibles et pour quels environnements (quels fonds d'image sur lequel la cible devra être extraite et suivie) on peut utiliser chaque méthode.

Méthode	Sorte des Cibles	Environnement
Contours actifs	Objets ou personnes de forme connues ou inconnues, avec contours définis par forts gradients ou avec zones uniformes soit en couleur, soit en texture.	Peu ou pas du tout encombré, mais avec possibilités de changements importants d'illumination
Corrélation 1D	Lignes ou objets structurés (typiquement, polyédriques), principalement sans texture.	Selon la complexité de la cible, de pas du tout à un peu encombré. Tolère de légers changements d'illumination.
Ensemble de points	Zones a priori non connues, mais, assez texturées pour extraire des points d'intérêt, de contour.... Zone de recherche contrainte par le type de mouvement de la cible, s'il est défini	Peu ou pas encombré, avec légers changements d'illumination.
Différences de motifs	Zones planes ou quasi-planes, texturées, a priori connues pour garantir leur planitude et éviter l'apprentissage en ligne.	Environnement quelconque, avec légers ou presque nuls changements d'illumination

Table 3.2. Cibles et environnements pour chaque méthode.

Comme on peut voir dans la table 3.2, montrant les caractéristiques de chaque méthode, les cibles sont aussi variées que les méthodes. De manière évidente, les méthodes basées sur les contours ou les régions respectivement, ont des comportements différents ; tandis que les unes (corrélations et contours actifs) suivent les bords d'une cible, donc fonctionnent d'autant mieux que ce bord se détache (fond uniforme, cible non texturé), les autres (ensemble de points, différence de motifs) suivent des caractéristiques à l'intérieur de la cible, donc nécessitent une cible texturée et s'accommodent mieux d'un fond quelconque.

En principe, avec la méthode de contours actifs il est possible de suivre toutes sortes d'objets. Cependant, le type d'environnement restreint l'usage de cette méthode, puisque les *snakes* non contraints sont très sensibles aux contours présents en fond d'images. Contrairement aux autres méthodes, avec la restriction précédente près, il est possible de suivre des objets avec des conditions extrêmes d'illumination; cela est dû à ce que les équations de mouvement donneront toujours une solution, même avec un minimum de potentiel externe.

Pour le suivi par corrélation 1D, il faut que la cible ait, une bonne définition de contours, et même s'il est possible de suivre un objet complexe dans un environnement un peu encombré, la méthode dépend directement des extracteurs de caractéristiques à

corréler et donc, les changements d'illumination peuvent perturber les résultats significativement.

De manière similaire à la méthode de corrélation 1D, la méthode basée sur la distance de Hausdorff, récupère les points de l'ensemble par moyen d'un détecteur des caractéristiques, et de ce fait, dépend aussi des changements en illumination.

Finalement, la méthode de différences de motifs, peut traiter des cibles très complexes dans presque toutes les conditions d'environnements, à condition quelle soient planes ou quasi-planes. Il est possible seulement de traiter des petits changements d'illumination, principalement dû au fait que l'apprentissage de la cible a été fait sous certaines conditions.

3.3.3 Caractéristiques des méthodes et détection d'erreurs.

Reprenons dans la table suivante, les caractéristiques de chacune des méthodes ainsi que les possibilités internes de détection des erreurs. A ces critères de détection interne, se rajoutent les mesures externes liées aux brusques variations de la représentation de la cible entre deux itérations du suivi.

	Contours actifs	Corrélation 1D	Ensemble de points	Différences de motifs
Définition de la cible	Contours externes, silhouette	Contours ou bords de la cible	Région plutôt texturée	Région plane texturée
Espace d'observation	Gradients des caractéristiques désirées	Corrélations entre caractéristiques	Appariement de points dans la région	Différences de motifs
Génération des hypothèses	Evolution du contour à partir de l'équation de mouvement ou par programmation dynamique	Ensemble aléatoire généré à partir des points de corrélation maximale	Recherche de la cible dans une zone définie à partir du modèle de mouvement	Simple calcul d'un vecteur de mouvement de la cible, grâce à la matrice d'interaction apprise off line
Evaluation des hypothèses	Minimum d'énergie	RANSAC	Distance de Hausdorff	Différences des motifs
Détection d'erreur interne à la méthode.	Non	Seuil sur les points corrélés et nombre maximal des hypothèses générées	Seuil sur le nombre de points appariés et sur la taille de la zone de recherche.	Seuils par rapport à l'apprentissage de la cible : score SSD et variations maximales apprises.

Table 3.3. Principales caractéristiques de chaque méthode.

3.3.4 Avantages et inconvénients

Pour finaliser la comparaison des méthodes dans la table 3.4, on donne un résumé des avantages et des inconvénients les plus significatifs de chacune des méthodes de suivi.

Méthode	Avantages	Inconvénients
Contours actifs	Méthode rapide. On peut suivre des objets déformables a priori inconnues, ainsi que des objets structurés (avec contraintes de forme sur le <i>snake</i>)	Nécessite des objets avec hauts contrastes cible/fond, donc des environnements simples Presque impossible de retrouver une cible perdue.
Corrélation 1D	Très robuste en milieu structuré	Très sensible dans des conditions d'environnements complexes
Ensemble de points	Peut traiter des objets plans ou 3D ou déformables, si les changements de pose sont graduels ; Peut retrouver des cibles perdues en élargissant la zone de recherche	Méthode lente. Des environnements complexes (fond texturé, nombreux points de contour) peuvent perturber la localisation
Différences De motifs	Très rapide. Robuste, même dans des environnements complexes. Il est possible d'utiliser différents modèles de mouvements (projectif, affine...)	Seulement pour des cibles planes ou sous certaines conditions, des cibles quasi-planes. Une fois que la cible est perdue, presque impossible de la retrouver, sauf pour des occultations partielles.

Table 3.4. Avantages et inconvénients des méthodes de suivi.

3.4 Analyse de performances pour différentes contextes

Afin d'analyser la réponse de chacune des méthodes pour différentes conditions de cibles et des environnements, on peut caractériser, autant les environnements que les cibles, par une mesure de leur complexité. Par exemple, la complexité de l'environnement ou de la cible, peut être une fonction des conditions d'illumination et/ou de l'encombrement et/ou de la régularité des formes qu'il contient, ce qui peut se traduire par un critère sur la texture et/ou les couleurs présentes, etc. Même s'il existe plusieurs mesures pour caractériser cette complexité, une seule mesure ne suffit pas, et donc il faut généralement en considérer plusieurs.

Malheureusement, les mesures de complexité dans une image, sont des mesures qui dépendent du contexte ainsi que de la tâche à accomplir. Déjà la sélection *automatique* des cibles en fonction de *bonnes caractéristiques*, vis-à-vis d'une application donnée, pour accomplir une tâche ou un objectif donné dans un environnement spécifique est toujours un problème [9], [16], [97].

Vu la quantité de variables de chacune des méthodes et les différentes formes de représentations de la cible pour chacune, une analyse statistique, au sens formel, des performances, semble une tâche extrêmement compliquée. La plupart des analyses de performances qui existent dans la littérature sont réduites à certaines conditions et pour certains jeux de paramètres, et en plus pour la plupart d'entre elles, l'analyse est faite pour des méthodes de suivi similaires, par exemple pour des méthodes avec la même représentation de la cible, ou avec le même espace de générations des hypothèses. Par exemple dans [38] Trois variantes de la méthode de suivi à partir de différences de motifs, ont été analysées et comparées, donc avec la même représentation de la cible. Ces méthodes ont été analysées en fonction des déformations, comme des translations, rotations et changements d'échelle. Notons que cette analyse a été faite pour certaines cibles, et qu'une analyse de la complexité ou de la texture des cibles ou de leur environnement n'est pas prise en compte. Comme nous le montrerons ce choix peut être crucial.

Mentionnons d'autres études de performances entre toutes celles qui ont été publiées ; dans [14], [41] et [104] ont été analysées différentes méthodes de suivi pour l'interaction Homme-Machine. De manière similaire à l'analyse décrite précédemment, les méthodes utilisent soit la même représentation de la cible, soit le même espace de génération des hypothèses. En plus, dans la plupart des analyses de performances des méthodes de suivi, les auteurs considèrent qu'il est possible de connaître automatiquement et avec une très bonne exactitude la localisation de la cible dans l'image, tâche qui à nouveau dépend du contexte ; dans la plupart des cas, elle a été résolue, par des différences entre une image de base, sans la cible à suivre et les images qui contiennent la cible. Malheureusement, cette approche n'est valable que pour des caméras fixes ou semi fixes, et non pour des caméras qui se déplacent sur un robot.

Par la suite, nous ferons une analyse des performances **non exhaustive**, de chacune des méthodes en prenant en compte : a) leur environnement, sa complexité, et les changements d'illumination, et b) les déformations de la cible dues aux rotations, translations et changements d'échelle. Cette analyse, sera par la suite utilisée pour caractériser les cibles et les conditions pour lesquelles chacune des méthodes donne de bons résultats.

A terme, les résultats pourront être améliorés, avec l'expérience acquise par le robot dans certains environnements et sous différentes conditions, comme des êtres humains le feraient. Bien sûr, il y a encore du chemin à parcourir pour ce faire ; aussi, pour l'instant nous considérons que c'est un expert qui définit les cibles pour le test, et qui évalue la complexité de l'environnement.

L'évaluation de performances a été faite dans différents contextes et conditions (figure 3.8), ainsi que pour différentes cibles. Cependant comme il a été mentionné dans le chapitre 1, des objets quadrangulaires et plans, comme des affiches ou des panneaux de signalisation, sont une bonne source d'information, pour la navigation dans un milieu humain. C'est pourquoi une grande partie des images acquises pour l'évaluation utilisent cette sorte de cibles.

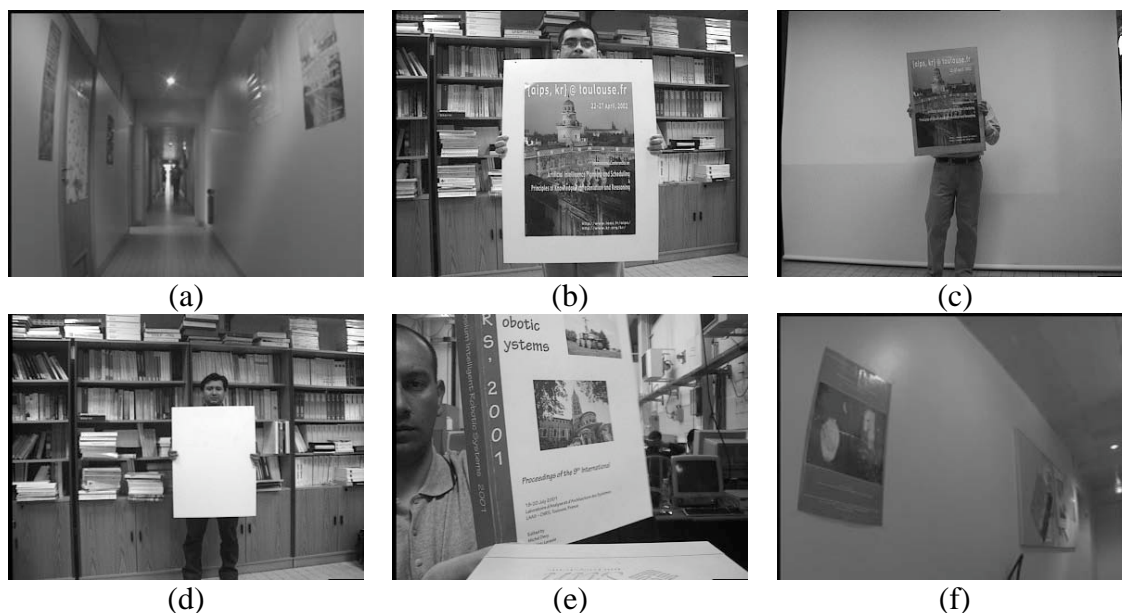


Figure 3.8. Exemples de différents environnements utilisés, pour l'évaluation de performances.

3.4.1 Caractérisation des contextes pour l'évaluation des performances.

La caractérisation de l'environnement, commence par la mesure de sa complexité. Différentes mesures ont été proposées dans la littérature. Cependant comme mentionné dans la section précédente, la mesure de la complexité dépend directement de la tâche à réaliser et bien sûr du contexte. Malgré cela, nous avons essayé, à partir de notre expérience personnelle, de trouver une mesure qui soit suffisamment robuste pour certains contextes connus ou possibles dans lequel un robot peut se trouver durant ses déplacements, et pour des tâches bien définies, comme la localisation par amers dans la navigation.

Les différentes mesures proposées dans la littérature sont très nombreuses, et comme le champ de développement des mesures de complexité contextuelle d'une image dans la communauté Vision par Ordinateur, est encore en évolution, il est très difficile d'établir une classification entre ces propositions.

Par exemple, il existe des mesures de la complexité pour une image binaire, comme par exemple la longueur du bord entre les zones blanches et noirs, mesures généralisées pour les images en niveaux de gris, en considérant chaque bit à l'intérieur de l'octet qui code l'intensité lumineuse. C'est à partir de la combinaison de ces mesures (par exemple, huit pour une image de niveaux de gris) qu'on peut avoir une mesure de la complexité de l'image entière [86]. Cependant, c'est un processus laborieux, coûteux en temps, et qui finalement, fournit des résultats comparables à des autres méthodes.

Une autre approche intéressante décrite dans [66] consiste à mesurer la complexité en fonction des mesures de l'information, tel que l'entropie, la température et l'harmonie, ou comme combinaison de ces mesures, ou encore par le nombre d'objets présents dans la scène, extraits par une méthode de segmentation.

Dans [89], différentes mesures de complexité d'images, ont été analysées pour la reconnaissance automatique d'objets, et puis réparties dans deux grand groupes : les mesures qui dépendent des objets recherchés dans les images et celles qui sont indépendantes. Pour le dernier groupe ces mesures peuvent être globales ou régionales, et pour le premier seulement régionales.

Ces méthodes peuvent aussi être classées en fonction de l'information extraite des images, comme par exemple, niveau de gris, contours, et dans le cas de mesures dépendantes des objets, forme et/ou taille de ceux-ci.

Par exemple, mentionnons quelques mesures globales sur l'image :

- en fonction des niveaux de gris, on peut citer : la déviation standard des niveaux de gris, l'entropie de l'image, l'uniformité, le ratio signal sur bruit, etc.
- en fonction des contours : le nombre de contours par unité de surface, la distance des pixels au contour voisin, etc.

et quelques mesures des régions :

- en fonction de niveaux de gris : l'uniformité de la région, les contrastes locaux, l'entropie structurale, etc.
- en fonction de contours : l'intensité du gradient le long des lignes, ou la longueur de bords des régions, etc.

Nous avons testé plusieurs mesures parmi celles-ci ; nous avons considéré qu'en fonction de cette expertise, pour les contextes analysés, une mesure de région en fonction de la « température » de l'image est un bon descripteur de sa complexité. Cette mesure n'est que la densité de variations d'intensité lumineuse par unité de surface (figure 3.9). La complexité d'une région donnée est donnée par la moyenne des valeurs de la température dans cette région.

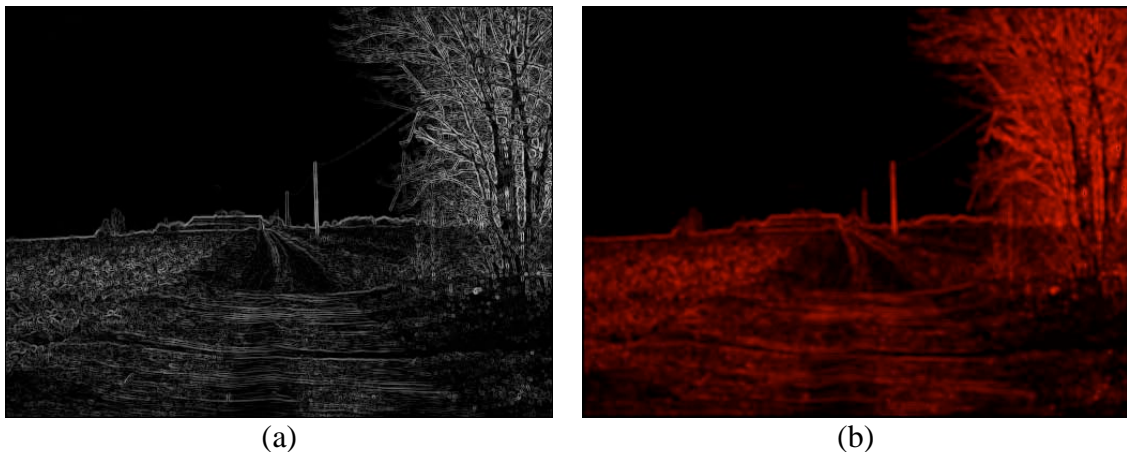


Figure 3.9. Densité des variations lumineuses par unité de surface : a) image de gradients d'intensité, b) image de la température spatiale, température moyenne égal à 35.

3.4.2 Caractérisation de cibles et d'environnements.

Afin de réaliser les mesures de complexité par région et non globalement sur l'image entière, trois zones ont été définies à l'aide de rectangles qui entourent la cible. Ces zones sont :

- a) la *zone des contours*, qui est une petite zone, entre 5 à 10 pixels d'épaisseur autour des contours de la cible,
- b) la *zone cible*, qui est toute la zone à l'intérieur du contour, et
- c) la *zone extérieur*, qui est une zone à l'extérieur du contour, qui s'étend jusqu'à la limite de la zone de recherche de la cible dans l'image, typiquement entre 10 à 20 pixels du contour de la cible.

Nous avons mesuré la température pour chaque zone afin de les caractériser, et par la suite classer les sortes de cibles pour lesquels chaque méthode de suivi fonctionne ou donne de bons résultats.

Dans les images de la figure 3.10, nous montrons des cibles perçues dans des contextes différents et les mesures de température correspondantes, taille de la zone du contour est de 8 pixels, et la zone à l'extérieur est de 20 pixels.

Dans les images (a), (b), (c) et (d), sont montrées quatre cibles rectangulaires planes. La région considérée pour l'analyse est comprise à l'intérieur du rectangle extérieur (rouge), pendant que la cible est le rectangle intérieur (vert).

Dans (a), la cible est texturée et a une zone de couleur homogène autour d'elle, la région d'analyse s'étend jusqu'à la limite de cette zone, avec quelques risques de perturbations avec cette limite. Dans (b), il n'y pas de zone de couleur homogène autour de la cible ; dans (c), la cible est entourée par une zone de couleur homogène plus grande que la région. Finalement, dans (d) la cible et la région sont dans une zone de couleur homogène, mais avec des variations d'illumination, principalement dues à la réflexion dans la surface de la cible.

Les images (e), (f), (g) et (h), montrent les variations d'intensité lumineuse ou gradients (*température*), des zones sélectionnées. En noir sont les gradients qui appartiennent à la *zone contour*, en vert ceux que appartiennent à la *zone cible*, et en rouge ceux qui sont dans *zone extérieur*.

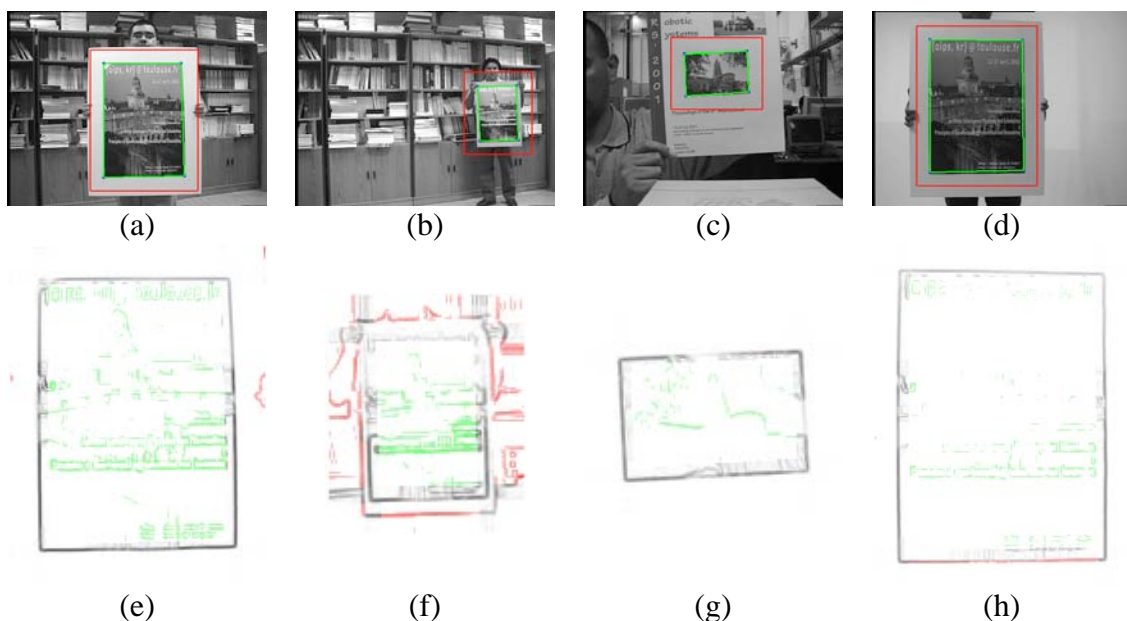


Figure 3.10. Différentes cibles dans différentes conditions d'entourage et quantité de gradients par zone d'intérêt.

La table 3.5 montre les valeurs obtenues de températures pour ces zones. Des zones encombrées ou texturées apparaissent avec une température élevée.

Zones	Cible			
	(a)	(b)	(c)	(d)
Contour	24.93	29.47	18.99	18.43
Cible	20.84	29.44	17.71	12.73
Extérieur	1.85	26.22	1.06	2.76

Table 3.5. Valeurs de moyenne de variations d'intensité lumineuse pour les différentes zones comprises.

Comme cela était prévisible par la théorie, autant les contours actifs que le suivi de lignes par corrélation 1D, fonctionnent bien quand la température de la zone des contours est plus forte que celles des zone intérieur et extérieur de la cible. Dans les cas contraires, le recalage des hypothèses sur la cible peut être perturbé par ces zones.

Pour les méthodes de suivi de régions (différences de motifs et ensemble de points), tant qu'il existe une bonne distribution de gradients autant à l'intérieur de la cible comme dans son contour, elles donneront de bons résultats. Même, dans certaines conditions il est possible de suivre des objets avec une température extérieure élevée.

Cependant, pour la méthode basée sur la différence de motifs, comme une partie de la zone à l'extérieur de la cible est prise en compte lors de l'apprentissage, quand cette zone change, on risque de perdre la cible ou simplement d'avoir des faux résultats. Similairement pour le suivi d'un ensemble de points, des changements de la partie extérieure à la cible peuvent provoquer des modifications du modèle de la cible par incorporation de ces caractéristiques qui n'en font pas partie initialement. Cela provoque soit la perte de la cible, soit son altération, ce qui fait qu'elle peut rester collée ensuite sur des caractéristiques extérieures.

Dans les figures 3.11a et b, sont montrées deux images avec la même cible mais avec des changements d'illumination, et dans les figures 3.11c et d, sont montrées les images de gradients extraits sur les zones d'analyse. Comme on peut le constater, la variation des gradients est importante, faisant que la méthode de différences de motifs ne marche pas. Par contre, soit avec les contours actifs, soit avec le suivi par corrélation 1D, il est possible de suivre la cible, même avec des faibles gradients, bien sûr avec ces conditions d'environnement, c'est-à-dire peu encombré, donc fond uniforme. Le suivi avec la distance de Hausdorff est possible à condition que la variation soit suffisamment lente, pour permettre l'actualisation de la cible, et bien sûr à condition d'avoir une bonne quantité de points d'intérêt extraits à l'intérieur de la cible par le détecteur de Harris ou autre extracteur de points.

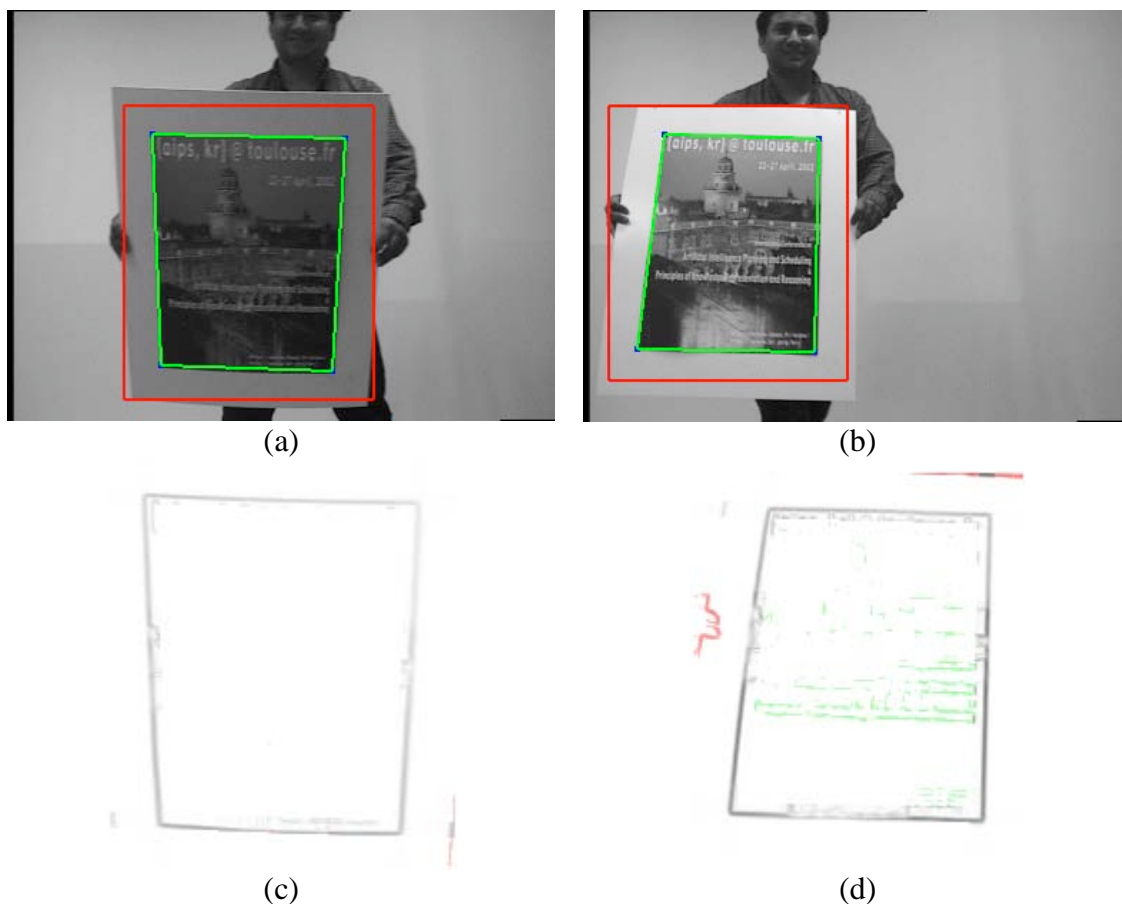


Figure 3.11. Changements d'illumination pour une même cible.

La figure 3.12 montre une cible, qui contient une bonne quantité d'information dans la zone d'analyse ; d'une certaine manière, cela peut être considéré comme un environnement structuré, à cause des lignes droites qui continuent dans le fond en dehors de la zone cible. Malheureusement, ce contexte pose problème :

- le manque de continuité dans les contours, fait qu'avec la méthode des contours actifs, le suivi n'est pas possible, du fait des facteurs de tension et rigidité, qui feront évoluer le contour vers l'intérieur de la cible.
- du fait de la structure présente dans l'environnement, c'est-à-dire des lignes droites dans le même direction que celles de la cible, la méthode de suivi avec corrélation 1D, connaît des problèmes pour le recalage des lignes sur les bords de la cible,
- il en est de même pour le suivi exploitant la distance de Hausdorff, car principalement, ces lignes seront incluses dans la cible au moment de l'actualisation de son modèle.
- la seule méthode qui marche dans ces conditions est la méthode de différences de motifs.

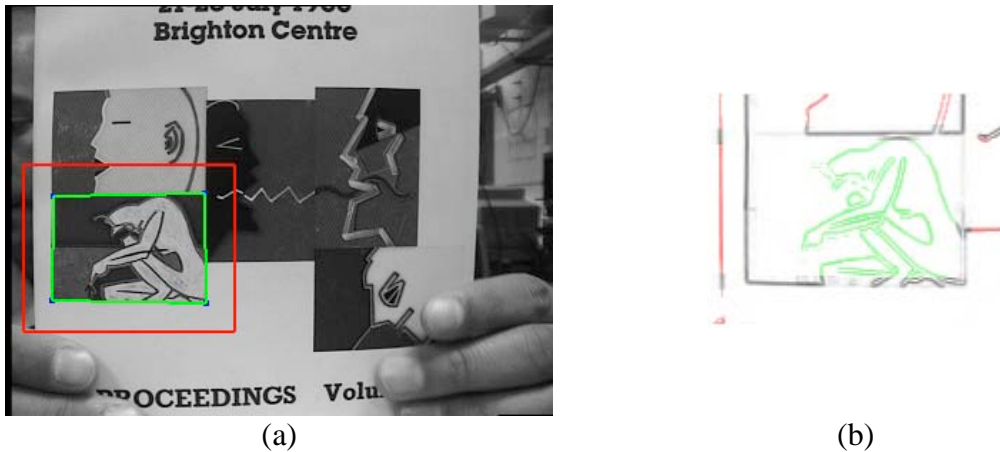


Figure 3.12. Cible dans un environnement relativement structuré.

Finalement, dans la table 3.6 nous donnons les résultats pour les différentes méthodes de suivi en fonction des mesures de la température dans les zones définies, où T_c est la température de la zone du contour, T_i est de la zone cible, et T_e de la zone extérieur.

Méthode de suivi	Conditions optimales	Commentaires
Contours actifs	$\frac{T_i}{T_c} < 0.75$ et $\frac{T_e}{T_c} < 0.75$	Donnes bons résultats même si la température T_c est très petite.
Corrélation 1D	$\frac{T_i}{T_c} < 0.68$, $\frac{T_e}{T_c} < 0.52$ et $T_c > 12$	Comme il a un seuil pour la corrélation maximal si $T_c < 12$, difficilement les points seront trouvés
Ensemble de points	$T_i > 10$, $T_c > 8$ et $\frac{T_e}{T_c} < 0.48$	Si la température dans les contours et à l'intérieur de la cible est inférieure à ces valeurs difficilement les points seront extraits
Différences de motifs	$\frac{T_i}{T_c} > 0.54$, $T_i > 14$ et $T_e \square cte$	La température T_i mesure en certaine manière la texture à l'intérieur de la cible, T_e peut avoir n'importe quelle valeur, a condition qu'il reste presque constant le long du suivi.

Table 3.6. Mesures et relations de températures des zones définies nécessaires pour le suivi avec chaque une des méthodes.

3.4.3 Déformations et variations de forme de la cible

Dans cette section, nous analyserons les performances de chacune des méthodes pour les variations de forme due aux changements d'échelle, rotations et translations. Afin de considérer des déformations projectives en plus des rotations autour de l'axe perpendiculaire au plan image (transformations affines), nous considérons les rotations autour des axes compris dans le plan image.

Pour ce test, nous adoptons pour chaque méthode, un contexte facile : bonnes conditions de la cible et de son environnement (figure 3.10f), bien que quelques perturbations –involontaires– aient pu survenir en dehors de la zone d'intérêt. Une séquence d'images a été acquise à une fréquence de 20Hz, afin de traiter avec les mêmes conditions toutes les méthodes ; dans cette séquence, les différentes conditions de déformations ont été considérées.

Dans un premier temps, un module d'acquisition d'images virtuel a été fait, qui assigne à chaque méthode de suivi, en fonction de leur fréquence de travail, l'image correspondante dans la séquence. Plus précisément on prend en compte le temps pris pour faire les calculs nécessaires pour chaque méthode, temps qui peut être non constant. Ainsi, par exemple, la méthode par différences de motifs (20Hz) exploite toutes les images de la séquence, tandis que la méthode par ensemble de points (3Hz), exploite une image sur 7 seulement.

Ensuite nous avons sélectionné des images similaires de la séquence pour faire la comparaison.

Les résultats sont les suivants :

- **Changements d'échelle.**

La figure 3.13, montre les résultats pour les variations d'échelle. Comme on peut le constater, toutes les méthodes donnent des résultats acceptables, sauf la méthode basée sur la distance de Hausdorff (en c et g), qui lors de l'actualisation du modèle de la cible, perd progressivement des informations sur celles-ci : le rectangle en lignes pointillées (rouge) est la zone de recherche et le rectangle en trait plein (vert) est la zone qui contient le modèle de la cible. On peut constater que le mécanisme prévu pour faire varier la taille de la cible, ne fonctionne pas correctement.

Pour les contours actifs (en b et f), on peut considérer que le suivi a été fait, même s'ils ont été déformés autour des coins supérieurs : les points sur l'image sont les points de contrôle du contour actif et la ligne continue, la courbe interpolée à partir de ces points de contrôle

Les figures a et e montre le suivi par différences de motifs, le contour pointillé (rouge) montre la prédiction précédente, et le contour continu (vert) la correction apportée après estimation du mouvement. Finalement, le suivi par corrélation 1D est montré dans les figures d et h où similairement aux différences de motifs le contour pointillé (rouge) montrent la prédiction et les lignes continues (vertes) montrent la correction.

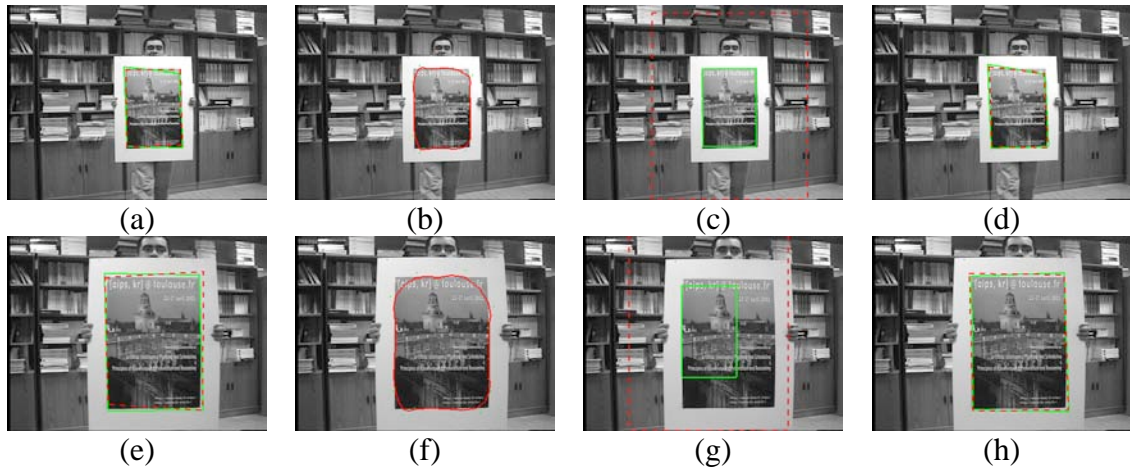


Figure 3.13. Analyse de performances pour des variations d'échelle.

- **Rotations.**

La figure 3.14 montre quelques résultats sur la sensibilité aux rotations de la cible. Les contours actifs présentent les mêmes problèmes de contraction dans les coins supérieurs, mais ici aussi, le suivi correspond bien à la cible. La méthode de Hausdorff est celle qui présente le plus de problèmes, du fait des rotations de la cible, principalement parce que la vitesse de rotation, fait que l'actualisation du modèle de la cible n'est pas fait à l'intérieur de la zone d'intérêt.

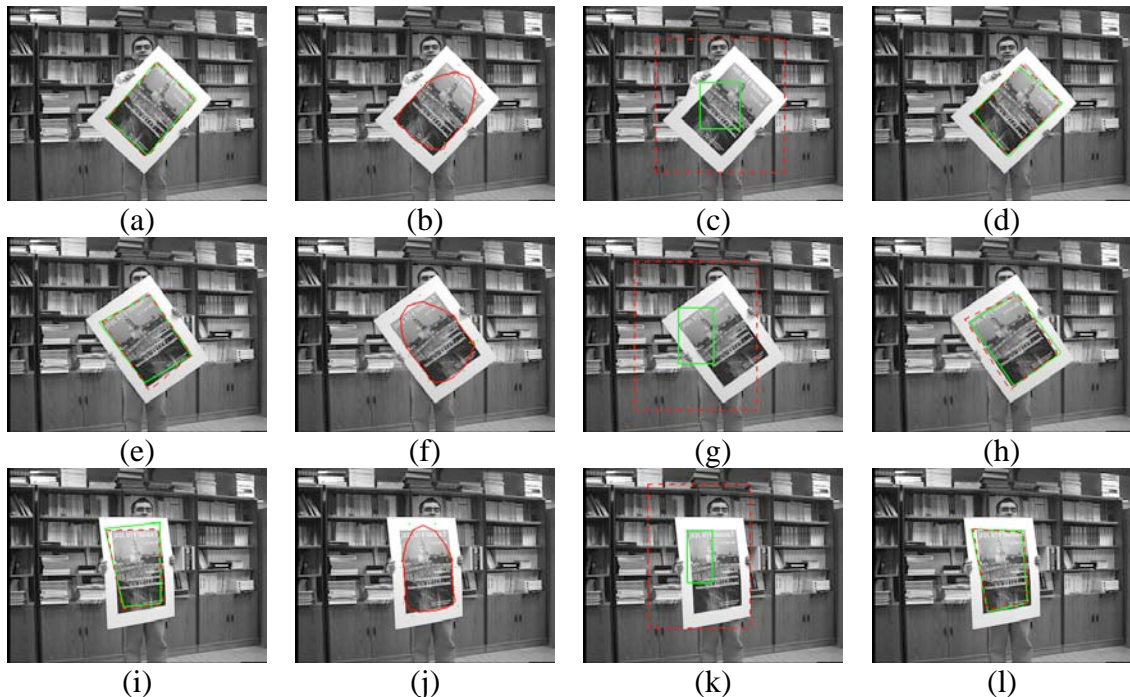


Figure 3.14. Rotations : de a) à h) rotations autour de l'axe perpendiculaire à l'image, et de i) à l) autour d'un axe parallèle aux colonnes de l'image.

- **Translations.**

Dans le cas des translations (figure 3.15), avec ces conditions la plupart des méthodes marchent, mais avec différents degrés d'exactitude sur le contour de la cible.

Seule le contour actif quelques fois reste collé pour certains intervalles de temps, soit sur des points à l'extérieur de la zone de recherche, soit à l'intérieur de la cible.

Par contre, dans des séquences où il y a plus de variations d'intensité dans la région d'intérêt comme cela est montré dans la figure 3.10b, autant le suivi par les contours actifs que le suivi par corrélation 1D auront des problèmes très importants pour suivre la cible.

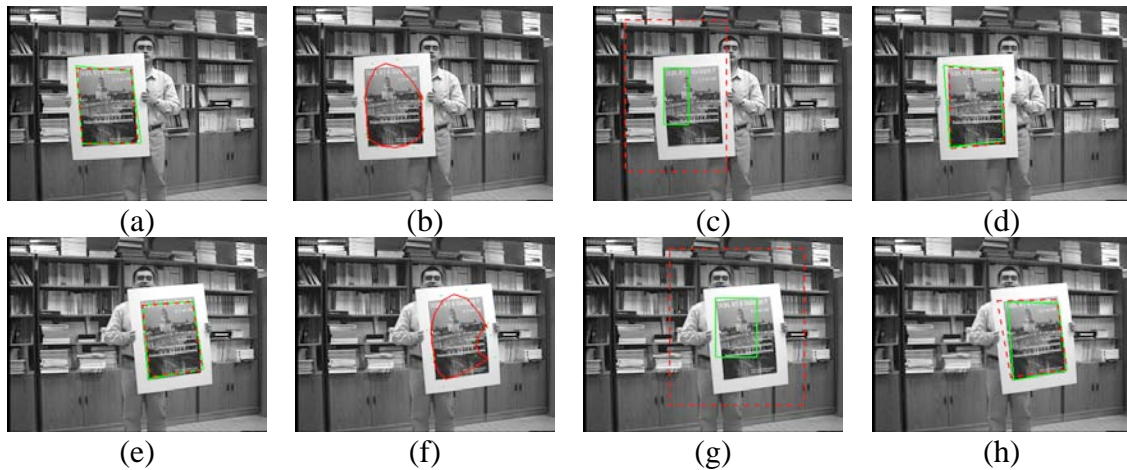


Figure 3.15. Translations

L'information obtenue dans cette analyse sera utilisée dans le chapitre suivant pour réaliser un plan perceptuel, et assigne à chaque cible la méthode ou méthodes que dans les conditions connues donne des bonnes résultats, ainsi que estimer, quand une erreur de suivi soit présente, soit essayer de reprendre l'activité avec les dernières bonnes résultats, et si ça n'est pas possible, donc essayer une autre méthode.

3.5 Conclusions

Dans ce chapitre nous avons caractérisé chacune des méthodes de suivi visuel en fonction des éléments suivantes : (1) la représentation de la cible, (2) la représentation de l'observation, (3) la génération des hypothèses et (4) la mesure des hypothèses.

Nous avons fait ensuite une comparaison des principales caractéristiques de chacune des méthodes en fonction de ces éléments, mais aussi de leur fréquence de travail, de la difficulté de les initialiser et de leur capacité de s'auto-évaluer, donc de détecter des erreurs. Cependant, comme les méthodes utilisent différentes formulations pour représenter les cibles, qu'elles ne s'appliquent pas toutes aux mêmes types de cibles, et comme, dans le contexte de la navigation, les conditions d'acquisition des images (mouvement du robot, de la caméra...), ne permettent pas l'extraction des positions réelles des cibles d'une manière facile, cette comparaison est restée qualitative.

Afin de caractériser les cibles et les conditions de leur environnement pour lesquelles chaque méthode donne de bons résultats, une mesure de température a été proposée, pour trois zones liées à la cible : (a) la zone du contour de la cible, (b) la zone à l'intérieur de la cible et (c) une zone à l'extérieur. Finalement, à partir de ces trois

mesures, nous avons étudié pour quelles combinaisons de température, chaque méthode a de bonnes performances. Enfin, une analyse de robustesse du suivi d'une cible, en fonction de mouvement apparent dans l'image (rotation, translation et changement d'échelle) a aussi été présentée.

L'information obtenue dans cette analyse des performances des méthodes de suivi, sera utilisée dans le chapitre suivant pour réaliser un plan perceptuel, et pour assigner à chaque cible, la méthode ou les méthodes qui dans des conditions contextuelles connues, donnent de bons résultats. Les mécanismes de détection d'erreur seront également exploités pour proposer des stratégies de reprise de la fonction à partir des derniers bons résultats.

Chapitre 4

Planification Perceptuel et Collaboration entre méthodes de suivi

4.1 Introduction

Dans la robotique, et plus précisément dans la robotique mobile, la planification est un des domaines les plus importants

On peut comprendre la planification comme la capacité de décider quelle ou quelles actions faire afin d'atteindre un but spécifique. Cette capacité permet à un robot de générer un *plan d'actions*, qui lui permettra de réaliser une tâche spécifique ; en général, le plan indique des pré-conditions et post-conditions pour l'exécution des actions, sans spécifier un ordre déterministe. Quand la tâche sera exécutée, les actions seront possiblement combinées d'une manière complètement non déterministe, comme prévu dans le plan ; c'est à partir de l'assemblage des capacités existantes de différentes manières que de nouvelles capacités peuvent émerger.

Nous imaginons un agent intelligent comme un système adaptatif qui puisse en complète autonomie, générer ses plans d'actions, et les modifier en cours d'exécution, en fonction de l'évolution de l'environnement. C'est à partir de la perception du monde qu'un robot pourra, soit évaluer l'état du plan actuel, soit modifier ce plan pour prendre en compte un événement non prévu, détecté par le système perceptuel. Citons un exemple classique :

- le planificateur de trajectoire exploite un modèle appris du monde dans lequel évolue le robot, afin de générer un plan de déplacement pour aller en un but donné (une courbe sur le sol, une séquence d'actions asservies...)

- le contrôle d'exécution est activé pendant l'exécution de ce plan, notamment pour détecter et éviter des obstacles non prévues, pendant les déplacements.

Entre tous les différents plans d'actions dont un robot peut avoir besoin, nous nous intéressons ici à la planification perceptuelle, à savoir, la génération d'un plan d'actions sur une caméra active montée sur un robot mobile, qui doit exécuter une trajectoire connue dans un contexte de navigation métrique. Nous avons repris et amélioré les travaux que V.Ayala Ramirez avait initiés.

Nous verrons ensuite comment contrôler l'exécution des commandes de suivi visuel présentées dans les chapitres précédents, et en particulier, comment gérer la coopération entre ces méthodes (enchaînement et reprise d'erreur).

4.2 Planification perceptuel

Le planificateur perceptuel qui nous intéresse, doit donner les configurations d'un capteur disposant de modalités contrôlables, nécessaires à un instant donné pour permettre au robot d'optimiser sa perception du monde. Nous sommes intéressés

- par le contrôle d'une caméra active, montée sur un robot mobile, dont on peut commander en ligne l'orientation de l'axe optique et le zoom,
- par la génération de plans perceptuels, dans le contexte de la navigation métrique : comment planifier les configurations d'une caméra active, pour optimiser la perception sur des amers visuels, exploités pour localiser le robot ?

D'abord nous allons définir les repères et variables utilisés.

Nous considérons le modèle suivant de la caméra (figure 4.1). A un instant donné t la caméra est à une position \mathbf{x}_t , avec une direction φ_t par rapport au repère global \mathbf{O}_{xyz} dans lequel est défini le modèle de l'environnement (monde). Si la caméra est montée sur un robot d'intérieur, elle se déplace sur un plan horizontal, parallèle au sol, donc au plan \mathbf{O}_{xy} .

Pour simplifier les notations, nous identifierons la position \mathbf{x}_t du robot et la position de la caméra dans le plan du sol \mathbf{O}_{xy} ; l'axe optique de la caméra sera orienté selon l'axe \mathbf{O}_y du repère robot lorsque les angles site et azimut sont nuls. Donc, dans cette configuration, l'orientation de l'axe optique est identique à l'orientation φ_t du robot. Par contre, nous supposons connue la hauteur h la caméra par rapport au plan du sol.

Par rapport à cette configuration $(\mathbf{x}_t, \varphi_t)$ dans l'espace, la caméra peut effectuer des rotations dans deux directions en décrivant une surface sphérique,

- dans la direction azimut, avec un angle φ (*pan*) avec $|\varphi| < \varphi_{\max}$ et
- dans la direction polaire, avec un angle θ (*tilt*) avec $|\theta| < \theta_{\max}$,

où φ_{\max} et θ_{\max} sont les butées maximales pour ces angles, ce qui limite les orientations possibles pour l'axe optique. Par définition l'angle polaire est défini comme l'angle entre le vecteur perpendiculaire au plan azimutal et la direction polaire. Nous pouvons, sans perte de généralité, le déplacer d'un angle $(\pi/2)$ afin de mesurer cet angle par rapport au plan azimutal (figure 4.1).

La caméra peut augmenter ou réduire son champ de vue par un angle $\varphi(\text{zoom})$, avec $\varphi_{\min} < \varphi < \varphi_{\max}$, où φ_{\min} et φ_{\max} sont les angles minimal et maximal respectivement pour l'ouverture du champ de vue dans le plan parallèle aux lignes de l'image.

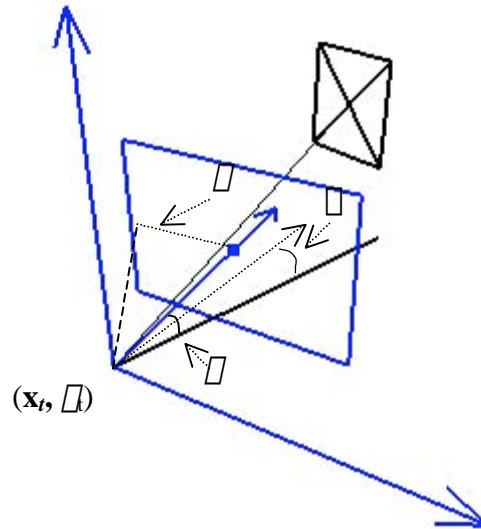


Figure 4.1. Modèle et paramètres de la caméra.

Une fois définis le modèle de la caméra utilisée, et ses paramètres contrôlables, on peut continuer avec les concepts, critères et mesures nécessaires afin de générer un plan perceptuel.

Comme mentionné au chapitre 1, pour la navigation métrique en milieu humain, les amers plans, comme des affiches ou des panneaux de signalisation (c'est-à-dire, des cibles quadrangulaires ou polygonales planes), sont courants dans cette sorte d'environnements. Ils sont une très bonne source d'information pour la localisation d'un robot mobile. C'est pourquoi nous nous focaliserons dans la création des plans perceptuels pour la navigation métrique, fondée sur une carte apprise hors ligne dans laquelle les amers plans sont positionnés dans le repère du monde.

L'approche proposée peut être étendue à n'importe quelle sorte des cibles ou d'amers.

Nous allons tout d'abord décrire des travaux qui pré existaient dans notre groupe de recherche ; puis nous verrons comment cette approche a été reprise, en tenant compte d'un critère de visibilité des amers et en proposant une méthode plus efficace afin de déterminer le plan perceptuel.

4.2.1 Approche initiale de planification pour la navigation métrique

Pour commencer, nous considérons un ensemble d'amers $\mathbf{L} = \{L_1, L_2, \dots, L_i, \dots, L_N\}$, connus a priori, répartis dans l'environnement, et une trajectoire générée par ailleurs, que le robot doit exécuter. Notre problème se résume ainsi : depuis un point \mathbf{x} de cette trajectoire, (1) quels amers le robot doit-il percevoir pour se localiser?, et (2) avec quelle configuration de la caméra ?

Des travaux précédents ont été faits au sein du groupe RIA, relatives à la planification perceptuelle pour la navigation métrique d'un robot mobile dans des environnements humains, plus précisément les travaux de V.Ayala-Ramirez [6] et JB.Hayet [51]. Nous commencerons par les décrire et les analyser.

Dans [6], la planification des modalités contrôlables d'une caméra active, est appliquée dans le contexte de la navigation basée sur des amers visuels pour un robot mobile devant se déplacer le long d'une trajectoire connue, dans un milieu intérieur. Une méthode a été proposée pour la sélection des amers les plus utiles en une position \mathbf{x} du robot ; cette méthode exploite une discrétisation de l'espace de configuration de la caméra et des grilles de visibilité et d'utilité associées à chaque point de cet espace discrétisé. La méthode a été mise en œuvre uniquement pour planifier l'orientation en azimut (φ) et l'angle du champ de vue ou zoom (θ).

Par la suite, nous présentons la manière proposée, pour le calcul de l'utilité pour les configurations du capteur, et le mécanisme de sélection de la configuration optimale (φ , θ) en une position \mathbf{x} de la caméra. La trajectoire pour le robot et donc, pour la caméra, est une courbe définie dans le plan du sol \mathbf{O}_{xy} , échantillonnée en un ensemble de positions ($\mathbf{x}_0, \dots, \mathbf{x}_t, \dots, \mathbf{x}_N$) ; en la position \mathbf{x}_t , le robot a la configuration (\mathbf{x}_t, φ), où l'orientation φ est donnée par la tangente à la courbe.

Le plan perceptuel est calculé hors ligne ; il fournit la sélection des configurations optimale pour chaque position \mathbf{x}_t le long de cette trajectoire. Ensuite, ce plan devra être exécuté sur le robot, en ligne ; un contrôleur devra alors traiter les situations non prévues dans la phase de planification (obstacles, occultations, non détection d'amers, pertes de suivi ...). Une phase de re planification en ligne pourrait être nécessaire, mais nous n'avons pas traité ce problème dans cette thèse.

Afin de sélectionner des modalités optimales de la caméra, en chaque position \mathbf{x}_t , une fonction d'utilité est évaluée pour chaque amer L_i : elle vaut 0 pour une utilité nulle, 1 pour une utilité optimale. Cette fonction est le produit de plusieurs termes, associés à différents critères d'utilité, qui auront une utilité égale à 1 quand le critère est optimal.

Décrivons les critères pris en compte dans l'approche initiale.

- **Visibilité de l'amer**

Le premier critère concerne la visibilité, car d'abord, il faut que l'amer soit visible depuis la position et la configuration courante de la caméra pour qu'il soit utile.

La visibilité géométrique $V_i(\varphi, \theta)$ depuis une configuration donnée $\mathbf{X} = (\mathbf{x}_t, \varphi)$ du robot et pour une configuration de la caméra active (φ, θ) est calculée comme :

$$V_i^{\mathbf{x}}(\varphi, \theta) = \begin{cases} 1 & \text{si l'amer } L_i \text{ est visible depuis la configuration actuelle} \\ \varphi & \text{caméra - robot} \\ 0 & \text{en dehors de cette zone de visibilité} \end{cases} \quad (4.1)$$

L'approche initiale ne traite pas les occultations : les amers sont supposés ponctuels dans un environnement vide.

En pratique, pour représenter tous les termes d'utilité pour chaque amer dans chacune des configurations $(\mathbf{x}_t, \vartheta_t)$ de la trajectoire du robot, V.Ayala propose de construire des grilles en discrétisant les angles ϑ et ϕ tous les degrés entre leurs valeurs minimales et maximales. Notons que ces grilles ne sont valables que pour une configuration du robot.

- **Taille de l'amer dans l'image.**

La taille de l'amer dans l'image, est directement liée au champ de vue ϑ , et à la distance \mathbf{d}_i de l'amer L_i au robot. Pour chaque amer, il est facile de calculer le meilleur angle de vue ϑ^* pour chaque distance amer-robot \mathbf{d}_i ; typiquement, connaissant la taille de l'amer dans le monde 3D, il s'agit de calculer ϑ^* pour que l'image de l'amer dans l'image ait une taille optimale. Le terme d'utilité pour cet amer L_i , et pour une configuration de la caméra (ϑ, ϕ) est noté $K_1^{L_i}(\vartheta, \phi)$; il donne l'influence de ce critère sur l'utilité globale de cet amer.

Un élément de la grille associée à ce critère reflète l'écart normalisé d'une configuration (ϑ_i, ϕ_i) de la caméra par rapport au meilleur champ de vue $\vartheta^*(L_i, \mathbf{X})$ quand le robot est dans la configuration \mathbf{X} .

$$K_1^{L_i}(\vartheta, \phi) = 1 - \frac{\|\vartheta - \vartheta^*\|}{\max_{\vartheta} \|\vartheta - \vartheta^*\|} \quad (4.2)$$

- **Alignement de l'amer par rapport à la trajectoire.**

Le terme d'utilité $K_2^{L_i}(\vartheta, \phi)$ de l'amer L_i prend en compte l'alignement de l'amer par rapport à la trajectoire du robot. Plus longtemps les amers peuvent être suivis, même si le robot change de direction, plus ils sont utiles. Par ailleurs, ce terme favorise les configurations pour lesquelles la caméra observe la scène devant le robot.

$$K_2^{L_i}(\vartheta, \phi) = 1 - \frac{\|\vartheta - \vartheta^*\|}{\max_{\vartheta} \|\vartheta - \vartheta^*\|} \quad (4.3)$$

avec ϑ^* , l'angle d'azimut que permettrait d'avoir l'amer L_i au centre de l'image pour l'itération suivante, donc pour la configuration $(\mathbf{x}_{t+1}, \vartheta_{t+1})$. Notons que le choix de ϑ^* permet une anticipation sur le prochain mouvement du robot, prochain mouvement connu du fait de la trajectoire.

De même, un élément de la grille associée à ce critère reflète l'écart normalisé d'une configuration (ϑ_i, ϕ_i) de la caméra par rapport au meilleur angle azimut $\vartheta^*(L_i, \mathbf{X})$, pour l'amer L_i quand le robot est dans la configuration \mathbf{X} .

- **Adéquation de l'amer pour la localisation.**

Si nous considérons des amers plans, fixés sur des plans verticaux, le point de vue optimal pour la localisation du robot, est d'avoir l'axe optique perpendiculaire au plan de l'amer. Si \vec{n} , désigne le vecteur normal à ce plan, dans le repère robot courant, l'utilité pour la localisation est maximisée si l'axe optique de la caméra est orientée dans la direction opposée. Un choix pour $K_3^{Li}(\vec{n}, \vec{d})$ est alors défini par: Adéquation de l'amer pour la localisation.

Si nous considérons des amers qui ne peuvent être visibles que dans la direction \vec{d} , l'utilité pour la localisation est maximisée si la caméra est orientée dans la direction complémentaire. Un choix pour $K_3^{Li}(\vec{n}, \vec{d})$ est alors défini par:

$$K_3^{Li}(\vec{n}, \vec{d}) = 1 - \frac{\|\vec{n} \cdot \vec{d}\|}{\|\vec{n}\| \|\vec{d}\|} \quad (4.4)$$

- **Fonction d'utilité globale**

La sélection de la modalité optimale pour la caméra active est obtenu à partir des grilles d'utilité globale $U_{Li}^{xi}(\vec{n}, \vec{d})$ construites pour chaque amer L_i . Cette grille globale fusionne les m critères d'utilité pour un amer L_i . L'expression pour calculer cette utilité globale est donnée par :

$$U_{Li}^{xi}(\vec{n}, \vec{d}) = \prod_{j=1}^m K_j^{Li}(\vec{n}, \vec{d}) \quad (4.5)$$

Pour l'ensemble des amers $L = \{L_1, L_2, \dots, L_N\}$, la configuration optimale pour la caméra $\mathbf{C}^* = (\vec{n}^*, \vec{d}^*)$ est le point (\vec{n}, \vec{d}) où l'utilité globale est maximisée:

$$\mathbf{C}^* = \max_{\mathbf{C}} \prod_{i=1}^N U_{Li}^{xi}(\vec{n}, \vec{d}) \quad (4.6)$$

Dans l'implémentation de cette approche par V.Ayala, le maximum de l'utilité globale, correspondant à la configuration optimale \mathbf{C}^* , est recherché dans la grille, somme de toutes les grilles construites pour les amers. Il s'agit donc d'une approche exhaustive, puisque pour toutes les positions sur la trajectoire, les utilités sont évaluées pour tous les amers et pour toutes les configurations de la caméra active. En figure 4.2, nous montrons une telle grille, construite dans l'espace discrétisée des configurations de la caméra.

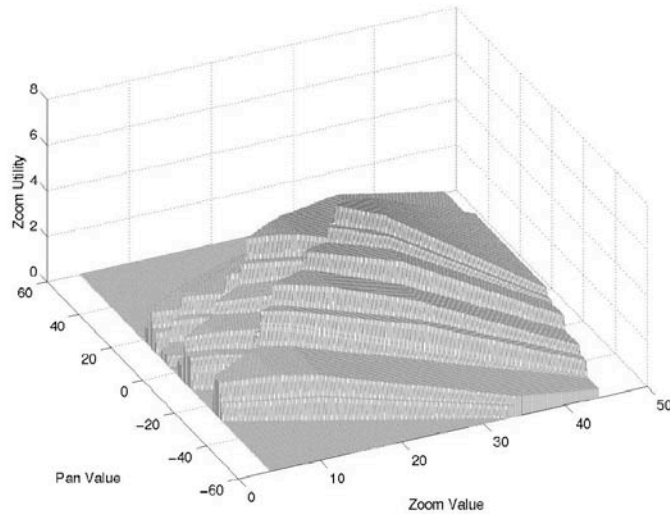


Figure 4.2. Grille d'utilité d'un ensemble d'amers pour une position donnée \mathbf{x}_t .

Notons quelques maladroites dans l'implémentation courante de cette approche :

- la notion de visibilité des amers ne prend pas en compte sa taille ni son orientation, ni les occultations, mais seulement sa position dans l'espace par rapport au plan \mathbf{O}_{xy} . Les considérations d'orientation et de visibilité partielle sont souhaitables afin d'éviter la recherche de l'utilité pour des cibles non visibles ou peu visibles à partir d'une position donnée \mathbf{x}_t ,
- Ce n'est que dans le calcul du critère K_3 que l'orientation de l'amer est considérée. De ce fait, la grille d'utilité d'un amer avec une orientation perpendiculaire à l'axe optique de la caméra sera calculée, même si cet amer n'est pas visible depuis la position courante de la caméra.

Comme la configuration optimale de la caméra est obtenue par un calcul exhaustif des critères d'utilité de tous les amers visibles, la prise en compte de l'angle polaire φ (tilt) va multiplier la complexité de l'approche par le nombre de valeurs échantillonnées pour cet angle, typiquement 60 en prenant une valeur par degré....

A la différence du travail décrit précédemment, dans [51], une étude plus complète de la visibilité a été faite. La visibilité y est définie en fonction des projections des 4 coins d'un amer plan, de la manière suivante :

$$V_i(\varphi) = \begin{cases} 1 & \text{si } \varphi(\varphi, \varphi) \in c(\{\mathbf{p}_{ik}(\mathbf{x}_t, \varphi, \varphi)\}_{k=1}^4) \\ 0 & \text{sinon} \end{cases} \quad (4.7)$$

où c est un prédicat constitué de contraintes unaires et binaires de la forme suivante.

$$c(\{p_{ij}\}) = \prod_{j=1}^4 ((u_{p_{ij}} > 0) \wedge (u_{p_{ij}} < d_u) \wedge (v_{p_{ij}} < d_v) \wedge (v_{p_{ij}} > 0)) \prod_{j,k \mid j \neq k} ((|u_{p_{ij}} - u_{p_{ik}}| > \square u_{\min}) \wedge (|v_{p_{ij}} - v_{p_{ik}}| > \square v_{\min})) \quad (4.8)$$

où (d_u, d_v) sont les dimensions de l'image, $(\square u_{\min}, \square v_{\min})$ les dimensions minimales de la projection de l'amer, et $(u_{p_{ij}}, v_{p_{ij}})$ les coordonnées dans l'image du coin p_{ij} . Le calcul de la visibilité est fait en fonction de l'intervalle de projection de ces points (figure 4.3) ; or, pour faire la projection de points dans le repère image il est nécessaire d'avoir les paramètres intrinsèques de la caméra, paramètres qui changent en fonction du champ de vue \square . Cette méthode est donc mal adaptée pour une caméra équipée d'un zoom.

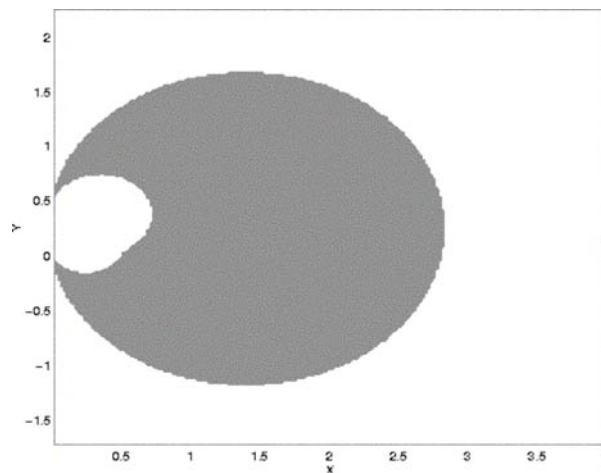


Figure 4.3. Visibilité d'un amer pour le plan xy , calculé en fonction des projections dans l'image de coins de l'amer, et des paramètres intrinsèques de la caméra.

Par contre, elle produit des cartes de visibilité des amers, partition du plan Oxy en régions ; pour une position \mathbf{x}_t , le robot est dans une région donnée, et c'est tel amer qui devra être perçu pour la localisation. Cependant, la carte de visibilité a été simplifiée pour ne retenir qu'un seul amer visible en toute position.

Nous allons par la suite, modifier la définition de visibilité ainsi que les critères d'utilité des amers, de manière à trouver une configuration optimale des paramètres de la caméra pour percevoir un ensemble d'amers visibles à partir d'une position donné \mathbf{x}_t , et cela sans faire appel à une recherche exhaustive. Subséquemment, nous allons optimiser la sélection d'amers et de paramètres de la caméra sur toute la trajectoire du robot, en introduisant aussi un terme fonction des variations des paramètres de la caméra entre deux positions successives de la trajectoire $(\mathbf{x}_t, \mathbf{x}_{t+1})$, terme nécessaire pour minimiser les c'est-à-dire en minimisant les saccades.

Nous commencerons par définir d'une manière simple une mesure, dans l'intervalle continue entre $[0,1]$, de la visibilité d'un amer L_i , en prenant en compte sa position dans l'espace, ses dimensions et son orientation.

4.2.2 Visibilité des amer plans

Nous sommes intéressé par définir la visibilité d'un amer donné L_i , en fonction de la position de celui-ci et de la caméra dans le repère du monde, en considérant d'une part ses dimensions et d'autre part, comme ces amers sont des cibles planes, son orientation donnée par le vecteur normal à la cible.

Prenons l'amer L_i , et soient, \mathbf{L}_i la position dans l'espace du centre géométrique de cet amer, \mathbf{N}_{L_i} le vecteur normal à cet amer tel que sa direction indique le coté visible de l'amer et \mathbf{x}_t la position d'un observateur (figure 4.4).

Définissons le vecteur \mathbf{d}_i comme :

$$\mathbf{d}_i = \mathbf{L}_i - \mathbf{x}_t \quad (4.9)$$

et l'angle φ_i tel que:

$$\cos \varphi_i = \frac{\mathbf{d}_i \cdot \mathbf{N}_{L_i}}{|\mathbf{d}_i| |\mathbf{N}_{L_i}|} \quad (4.10)$$

c'est-à-dire l'angle entre les vecteurs \mathbf{d}_i et \mathbf{N}_{L_i} .

La visibilité V_i de l'amer L_i depuis la position \mathbf{x}_t est définie comme :

$$V_i(\mathbf{x}_t) = \max(\cos(\varphi_i), 0) \quad (4.11)$$

L'équation (4.11) donne la visibilité V_i d'un amer *ponctuel* à partir d'une position donnée \mathbf{x}_t . Cependant, même si l'orientation de l'amer est prise en compte pour obtenir un intervalle de valeurs continues entre $[0,1]$, en lieu du *delta de Dirac* proposé dans [6], nous n'avons pris en compte ni la taille de l'amer, ni la distance par rapport à la position de l'observateur.

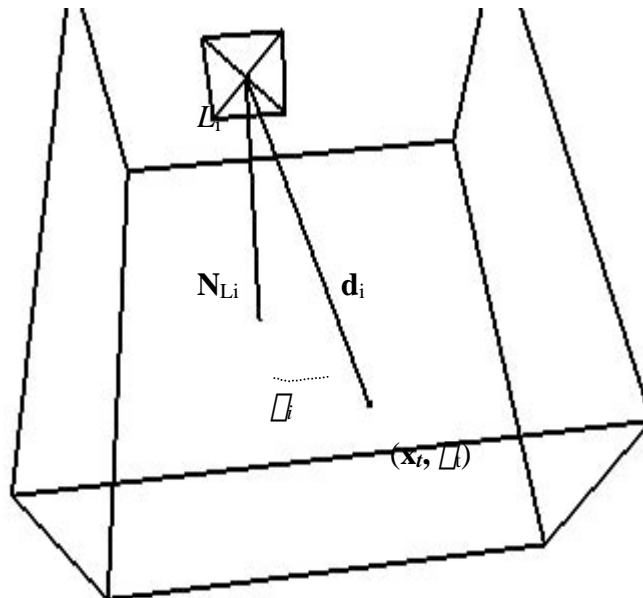


Figure 4.4. Visibilité d'un amer ponctuel.

La figure 4.5, montre à partir d'une projection sur le plan Oxy , les variations de la visibilité V pour deux amers (a) et (b) et pour différentes positions du point de vue.

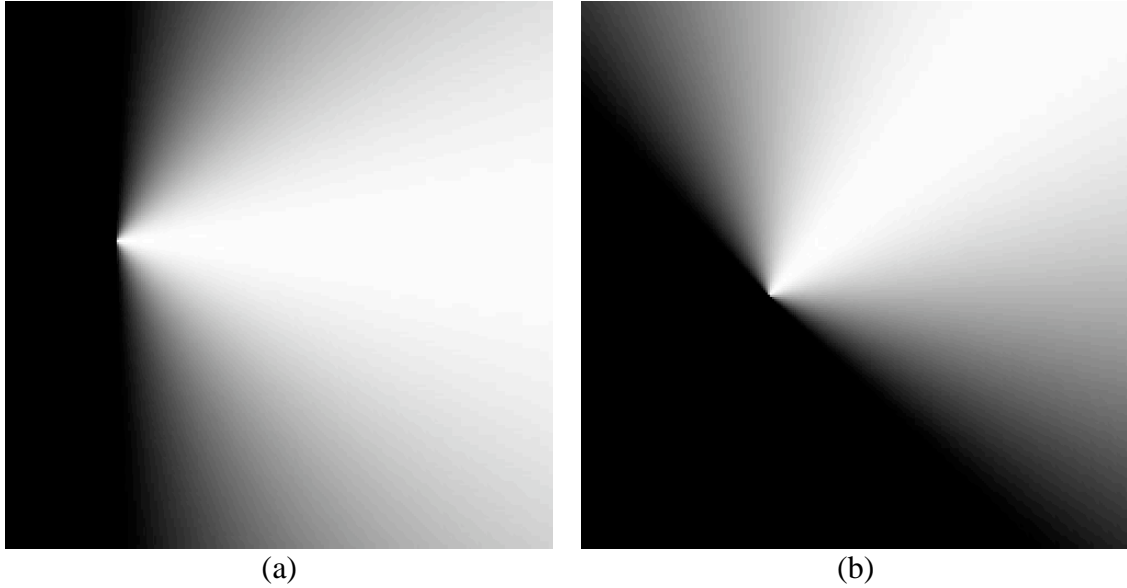


Figure 4.5. Carte de visibilité pour des amers ponctuels.

Afin d'éviter de considérer les amers seulement comme des points dans la définition précédente de la visibilité, on peut redéfinir la mesure de la visibilité en prenant en compte la taille de l'amer, et plus précisément ses coins.

Soient \mathbf{p}_{ij} , avec $1 < j < 4$, les quatre coins de l'amer i , comme cela est montré dans la figure 4.6. Définissons pour chaque point \mathbf{p}_{ij} , le vecteur \mathbf{d}_{ij} comme :

$$\mathbf{d}_{ij} = \mathbf{p}_{ij} - \mathbf{x}_t \quad (4.12)$$

et les angle \angle_{ij} , tel que:

$$\cos \angle_{ij} = \frac{\mathbf{d}_{ij} \cdot \mathbf{N}_{Li}}{|\mathbf{d}_{ij}| |\mathbf{N}_{Li}|} \quad (4.13)$$

Maintenant, la visibilité V_{Li} de l'amer i depuis la position \mathbf{x}_t sera définie comme :

$$V_{Li}(\mathbf{x}_t) = \frac{\sum_{j=1}^{n_c} V_{ij}}{n_c} \quad (4.14)$$

où n_c est le nombre de points pour lesquels la visibilité est calculée. Dans notre cas, en général, $n_c = 4$, et chaque V_{ij} est donné par :

$$V_{ij} = \begin{cases} \cos(\angle_{ij}) & \text{si } \angle_{ij} > \frac{\pi}{2} \\ 0 & \text{sinon} \end{cases} \quad (4.15)$$

La figure 4.7 montre la projection de deux amers sur le plan Oxy , et les valeurs correspondantes de la visibilité V pour différentes positions du point de vue.

Comme on peut le constater, à la différence de la définition précédente, quand sont considérées les dimensions de l'amer et les positions de ses coins dans l'espace, nous obtenons une nouvelle mesure de visibilité de ceci change, qui a un maximum en :

$$\lim_{|L_i \square x_r| \square \bullet} V_{Li} = 1 \quad (4.16)$$

et un minimum en :

$$\lim_{|L_i \square x_r| \square 0} V_{Li} = 0 \quad (4.17)$$

Ces équations permettent de conclure que quand on s'approche de la cible à une distance d très petite ($d \square 0$), sa visibilité sera presque nulle, ce qui est en accord avec notre notion de visibilité.

Par contre, pour obtenir une visibilité optimale, il faut s'éloigner sur l'axe normal à la cible ($d \square \bullet$), et l'optimum est obtenu à l'infini, ce qui paraît artificiel. Retournant à notre monde réel, où une distance infinie est impossible, on peut analyser les équations pour la visibilité et les calculer pour différentes cibles et positions. On s'aperçoit alors que pour une différence d'un ordre de magnitude entre la taille de la cible et la distance entre la caméra et celle-ci, sur la normale au plan cible, on arrive à une visibilité de 0.995 (figure 4.6). Ce qui veut dire que, pour des cibles de 10cm environ de côté, il suffit de distances d de l'ordre du mètre, pour avoir une bonne visibilité, selon notre modèle.

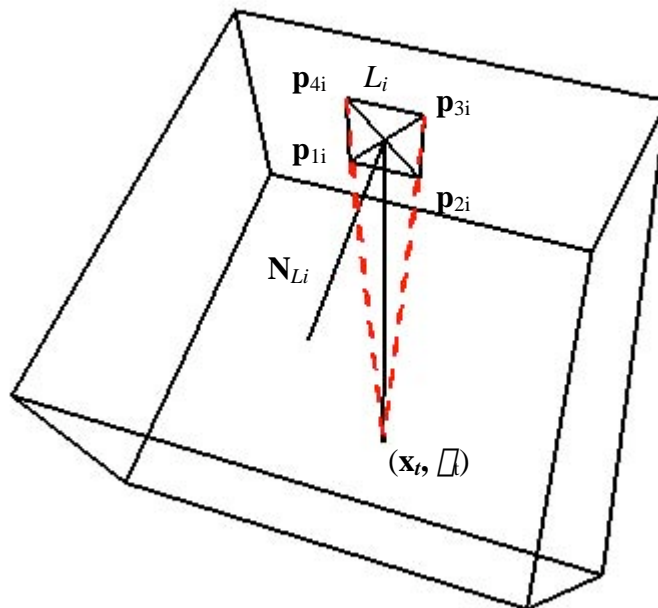


Figure 4.6. Visibilité d'un amer plan

La distance d est d'une certaine manière prise en compte pour le calcul de la visibilité dans l'équation (4.15). Notons que, dans ces équations, ni l'angle du champ de vue (zoom), ni la distance focale f de la caméra n'ont été pris en compte. C'est-à-dire, que notre notion de visibilité est pour l'instant indépendante des configurations possibles de la caméra. Ceci a été fait délibérément ; l'espace des configurations de la caméra sera pris en compte dans les définitions d'utilité que nous allons présenter.

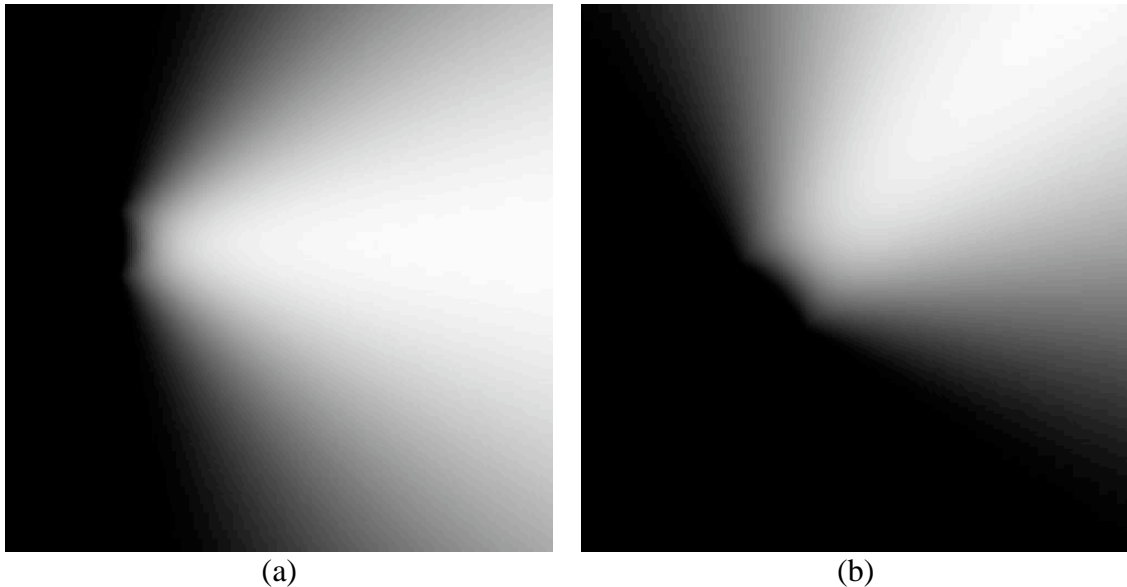


Figure 4.7. Carte de visibilité pour des amers plans de dimension l (indépendamment des paramètres de la caméra : focale ...)

4.2.3 Critères d'utilité.

Pour l'instant, nous avons défini la visibilité V_{Li} de l'amer i , depuis un point quelconque \mathbf{x}_t dans l'espace. Afin que cette définition prenne en compte les paramètres de la caméra, nous allons introduire ces paramètres, de manière similaire à l'approche présentée dans [6], c'est-à-dire, que nous allons définir des critères optimaux de point de vue en fonctions de ces paramètres. Pour cela, considérons d'abord les définitions suivantes :

Soit \mathbf{f} le vecteur focal de la caméra à la position \mathbf{x}_t , et rotation φ_t , tel que $f = |\mathbf{f}|$ est la distance focale, et soient \mathbf{u} et \mathbf{v} les vecteurs définissant le plan image (figure 4.8). Les vecteurs \mathbf{f} , \mathbf{u} et \mathbf{v} forment un système orthonormal appelé repère caméra, qui est fixe au point \mathbf{x}_t , mais qui peut subir deux rotations par rapport au repère caméra, avec les angles α et β (pan et tilt) définis précédemment.

Afin d'introduire le champ de vue, nous reprenons le critère K_1 défini par l'équation (4.2) ; nous considérons, comme proposé par Ayala-Ramirez [6], qu'il existe une valeur φ_i^* optimale du champ de vue, pour une distance donnée entre cible et caméra, distance donnée par le vecteur \mathbf{d}_i , vecteur entre les points \mathbf{x}_t et \mathbf{L}_i origine de la cible i . Cette valeur optimale est calculée en fonction des coins de l'amer, de manière similaire au calcul de la visibilité, pour que l'image de la cible ait une taille optimale en pixel, selon des contraintes liées à la méthode de suivi.

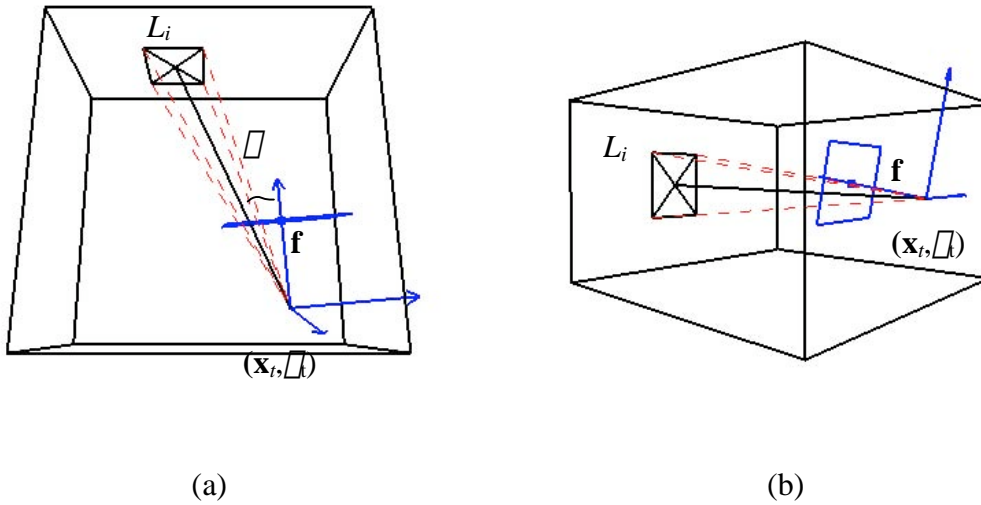


Figure 4.8. Ajustement de paramètres de pan et tilt

En suite, définissons à partir du repère caméra le critère de visualisation K_2 comme :

$$K_2^{L_i}(\mathbf{x}_t, \mathbf{f}) = \cos \square_i \quad (4.18)$$

où

$$\cos \square_i = \frac{\mathbf{f} \cdot \mathbf{d}_i}{\|\mathbf{f}\| \|\mathbf{d}_i\|} \quad (4.19)$$

et \mathbf{d}_i est toujours le vecteur entre les points \mathbf{x}_t et L_i , origine de la cible i .

Comme on peut le déduire de l'équation (4.19), les valeurs optimales de (\square, \square) , pour un amer donné seront celles qui aligneront le vecteur focal \mathbf{f} avec le vecteur \mathbf{d}_i , c'est-à-dire un angle $\square_i = 0$.

De manière similaire, nous pouvons définir un troisième critère en fonction de l'alignement de l'axe focal par rapport à la trajectoire du robot, comme :

$$K_c(\mathbf{x}_t, \mathbf{f}) = \cos \square \quad (4.20)$$

où

$$\cos \square = \frac{\mathbf{v} \cdot \mathbf{f}}{\|\mathbf{v}\| \|\mathbf{f}\|} \quad (4.21)$$

et \mathbf{v} est le vecteur de direction du robot dans la position \mathbf{x}_t (figure 4.9), vecteur dans le plan Oxy , d'orientation \square_t .

Cependant, comme on peut constater, plus qu'un critère dépendant des amers, c'est un critère qui dépend de la configuration optimale de la caméra, qui dépendra, comme nous le montrerons plus tard, de l'optimisation à partir des critères décrits précédemment (K_1, K_2), pour un ensemble d'amers visibles. C'est pourquoi nous affectons à ce critère l'indice c à la place d'un numéro 3.

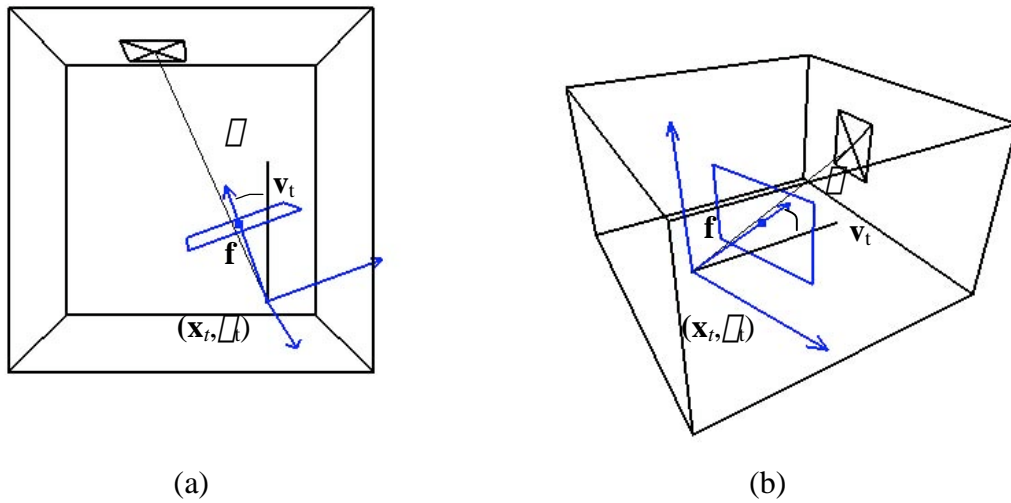


Figure 4.9. Angle entre le vecteur focal et la normal au robot.

Par la suite nous allons définir de manière similaire à l'équation (4.5), l'utilité d'un amer en fonction de sa visibilité et en fonction des critères dépendants de l'amer.

$$U_i^{xt} = \prod_{j=1}^m K_j^{Li} V_i(\mathbf{x}_t) \quad (4.22)$$

Définissons, l'utilité conjointe d'un ensemble de n amers visibles depuis la configuration $(\mathbf{x}_t, \varphi_t)$ de la manière suivante :

$$U_c^{xt} = \prod_{i=1}^n U_i^{xt} \quad (4.23)$$

Les paramètres optimaux de la caméra $(\varphi, \varphi, \varphi)$ doivent maximiser cette utilité conjointe, normalisée rappelons-le, entre 0 et 1.

Dans la figure 4.10, nous montrons les positions dans l'espace de trois amers plans orientés, ainsi que les normales à leurs plans support. La figure 4.11 montre la visibilité V_i de ces trois amers pour différentes positions de la caméra et pour différentes configurations azimut possibles (angle φ), pris dans l'intervalle $[-\varphi_{\max}, \varphi_{\max}]$. Dans cette figure, ni le critère K_1 , ni l'amplitude maximale du champ de vue φ_{\max} n'ont été pris en compte, c'est-à-dire, que pour chaque valeur du paramètre φ de la caméra on calcule l'utilité des trois cibles simplement comme le produit de la visibilité V_i , par le critère K_2 , et en considérant que toutes les cibles sont dans le champ de vue. Malgré cela, on peut considérer que dans la figure 4.11 sont représentés les composantes dans le plan Oxy , des mesures de visibilité des trois amers présentés dans la figure 4.10, depuis différentes positions \mathbf{x}_t . Comme on peut le constater, les zones de visibilité ne dépendent pas seulement de la position de la caméra dans le monde, comme il a été considéré dans [51], mais aussi de son orientation.

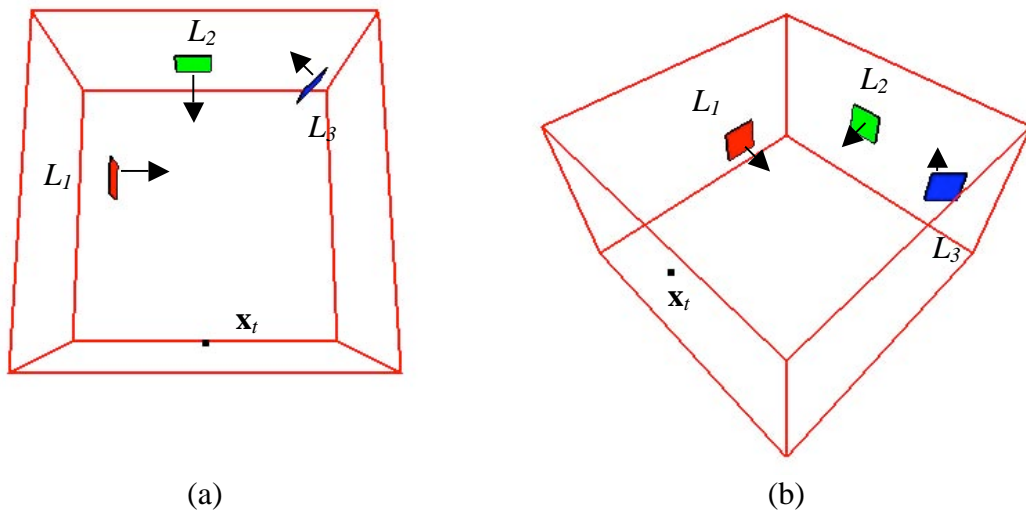


Figure 4.10. Configuration de trois amers plans dans l'espace

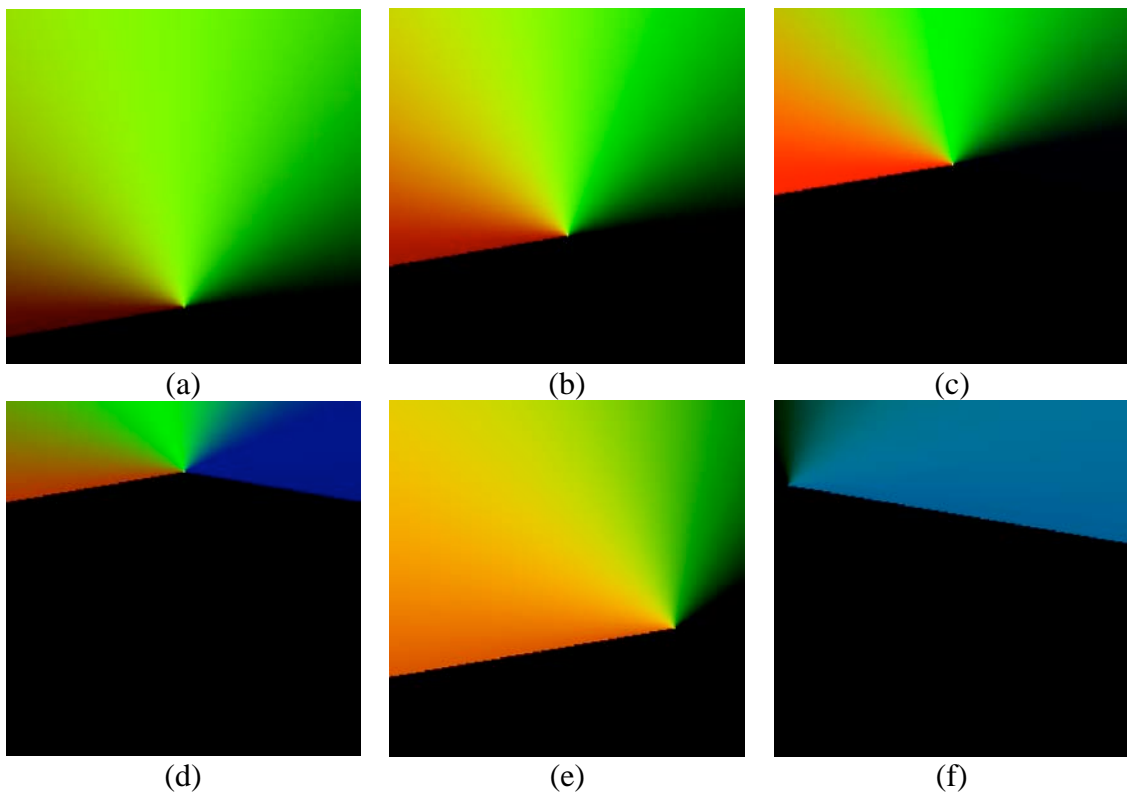


Figure 4.11. Visibilité de trois amers plans depuis différentes positions d'observation, les couleurs : rouge, vert et bleu représente la visibilité de chaque amer, donc la combinaison de couleurs donne la visibilité sur les trois amers pour une position donnée. Ici encore, les paramètres de la caméra ne sont pas pris en compte.

Comme mentionné, le facteur de champ de vue est très important afin d'évaluer l'utilité d'un amer, dans la (figure 4.12) nous montrons les changements de visibilité quand on considère les angles maximaux de champ de vue.

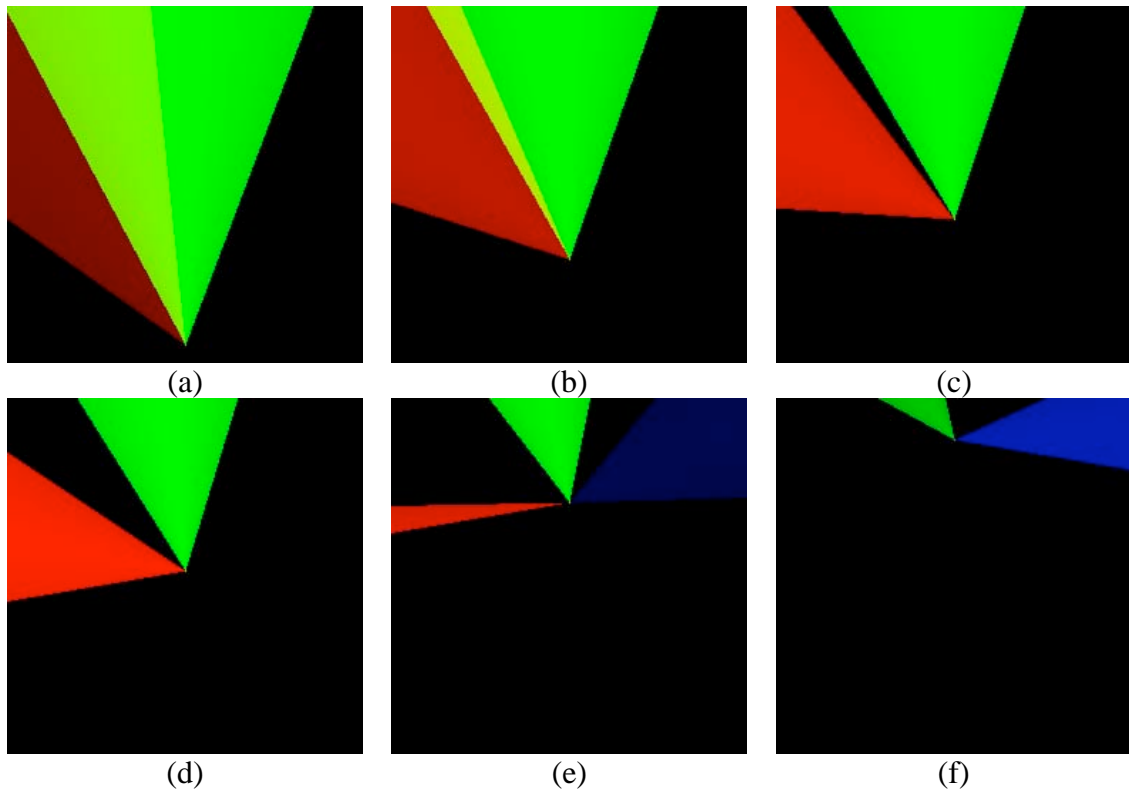


Figure 4.12. Visibilité des amers en considérant l'angle maximal d'ouverture du champ de vue.

A partir des figure 4.11 et de la figure 4.12, on peut voir, qu'une solution des valeurs optimales des paramètres de la caméra par une méthode exhaustive comme celle proposé dans [6], prend en compte de très nombreuses combinaisons de ceux-ci que en fait, ont une utilité nulle, ou très faible. Par ailleurs, au moment de prendre en compte l'angle polaire (tilt), la recherche devra être fait dans un espace de trois dimensions.

Par la suite, nous proposons une méthode analytique pour l'estimation des paramètres en fonction de valeurs de combinaisons des amers visibles depuis une position \mathbf{x}_i , ce qui permet d'éviter une recherche exhaustive.

D'abord, considérons comme amers utiles, depuis une position \mathbf{x}_i , ceux qui ont une valeur de visibilité plus grande qu'un seuil donné ($V_{Li} > V_{\min}$). A partir de ces n_v amers visibles, obtenons tous les k sous-groupes distincts d'amers visibles n_v .

Le nombre total de k sous-groupes distinctes est donne par :

$$\sum_{k=1}^{n_v} \binom{n_v}{k} = 2^{n_v} - 1 \quad (4.24)$$

parce que nous ne sommes pas intéressé par le sous-groupe nul.

En suite, il faut calculer à partir de l'équation de l'utilité conjointe, les paramètres pour chaque sous-ensemble d'amers. Pour cela on peut prendre en compte

deux sortes de sous-groupes d'amers visibles, ceux qui sont formés d'un seul élément, et ceux qui contiennent plus d'un amer.

Pour les ensembles formés d'un seul élément, le calcul des paramètres de la caméra s'obtient directement à partir de l'équation (4.23) :

- d'une part, on connaît la valeur optimale du champ de vue φ^* , comme il a été déjà mentionné pour le calcul de K_1 , et
- d'autre part, l'angle φ_i pour lequel l'utilité K_2 sera maximale, est celui qui donne la direction de \mathbf{d}_i , donc φ_i sera la combinaison des angles ϑ et φ par rapport au vecteur \mathbf{v} .

Pour les sous-groupes pour lesquels plus d'un amer sont visibles, c'est plus compliqué. D'abord il faut considérer, d'une part, que l'ensemble des paramètres de la caméra est le même pour tous les amers, et d'autre part, que l'angle ϑ apparaît seulement dans le calcul de l'utilité K_1 , alors que les angles ϑ et φ n'apparaissent pas directement dans le calcul K_2 .

On commencera par le calcul de ϑ . Pour cela, il est évident qu'il faut que les amers du sous-groupe soient tous dans le champ de vue de la caméra, donc l'angle de champ de vue sera ajusté sur la boîte englobante de tous les amers du sous-groupe, avec la condition $\vartheta < \vartheta_{\max}$. Si, cette condition n'est pas satisfaite, l'utilité de l'ensemble sera égale à zéro (nous pourrions ici considérer toutes les configurations d'amers visibles avec le champ maximal ϑ_{\max}).

Une fois trouvé l'angle ϑ qui satisfait $\vartheta < \vartheta_{\max}$, on peut considérer les facteurs V_{Li} et K_1^{Li} de l'équation d'utilité (4.22) comme constants, et donc il est possible de faire $C_i = K_1^{Li} V_{Li}$, et l'équation d'utilité devient :

$$U_i^x = C_i \cos \varphi_i \quad (4.25)$$

et l'équation (4.23) pour l'utilité conjointe du sous-groupe devient :

$$U_c^x = \prod_{i=1}^{n_v} C_i \cos \varphi_i \quad (4.26)$$

Considérons les angles suivants ϑ_i et φ_i , tel que ϑ_i soit l'angle azimut du vecteur \mathbf{d}_i par rapport au point \mathbf{x}_t et direction \mathbf{v}_t , et de manière similaire φ_i l'angle polaire entre \mathbf{d}_i et le vecteur \mathbf{v}_t au même point \mathbf{x}_t (figure 4.13).

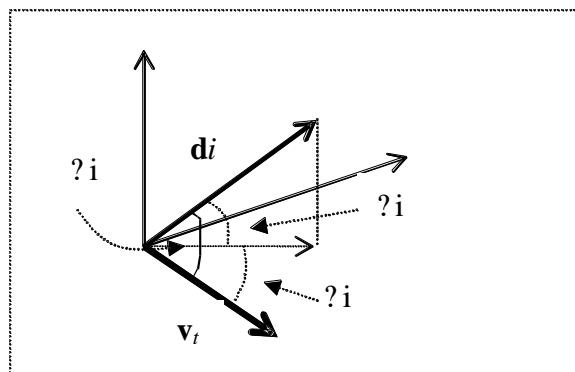


Figure 4.13. Angles azimut et polar entre le vecteur \mathbf{d}_i et le vecteur \mathbf{v}_t .

Sans perte de généralité on peut exprimer l'équation (4.26) en deux parties, de la manière suivante :

$$U_{\varphi}^x = \prod_{i=1}^{n_v} C_i \cos(\varphi_i - \varphi) \quad (4.27)$$

et

$$U_{\varphi}^y = \prod_{i=1}^{n_v} C_i \sin(\varphi_i - \varphi) \quad (4.28)$$

Prenons pour l'instant $n_v = 2$, et après quelques manipulations algébriques, nous obtenons le maximum de l'équation (4.28) pour :

$$\tan \varphi = \frac{C_1 \sin \varphi_1 + C_2 \sin \varphi_2}{C_1 \cos \varphi_1 + C_2 \cos \varphi_2} \quad (4.29)$$

et de manière similaire pour l'angle φ :

$$\tan \varphi = \frac{C_1 \sin \varphi_1 + C_2 \sin \varphi_2}{C_1 \cos \varphi_1 + C_2 \cos \varphi_2} \quad (4.30)$$

Généralisant pour $n_v > 2$, on obtient les équations suivantes :

$$\varphi = \tan^{-1} \frac{\prod_{i=1}^{n_v} C_i \sin \varphi_i}{\prod_{i=1}^{n_v} C_i \cos \varphi_i} \quad (4.31)$$

$$\varphi = \tan^{-1} \frac{\prod_{i=1}^{n_v} C_i \sin \varphi_i}{\prod_{i=1}^{n_v} C_i \cos \varphi_i} \quad (4.32)$$

Donc, à partir des équations (4.31) et (4.32), on peut estimer les paramètres optimaux φ et φ de la caméra pour k sous-groupes d'amers n_v , sans avoir besoin de faire une recherche exhaustive dans l'espace de configurations. Nous faisons par contre une recherche exhaustive dans l'ensemble des groupes d'amers visibles depuis la configuration courante $(\mathbf{x}_t, \varphi_t)$ sur la trajectoire du robot.

Pour résumer, dans la figure 4.14 nous montrons l'algorithme pour le calcul de l'utilité.

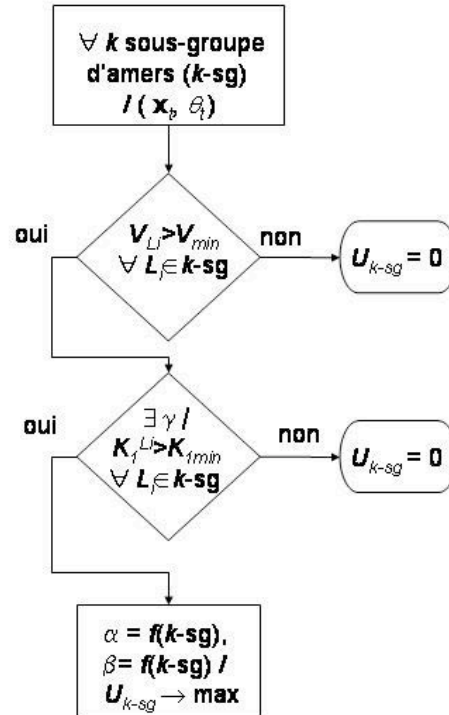


Figure 4.14. Algorithme pour estimer l'utilité d'un ensemble d'amers.

En ce moment on a $2^{n_v}-1$, mesures d'utilité pour la configuration courante (\mathbf{x}_i, θ_i) , c'est-à-dire une mesure pour chaque sous-ensemble. Donc, le sous-ensemble avec la plus haute mesure d'utilité peut être choisi, comme le plus approprié pour l'observation des amers depuis cette configuration.

Mais on peut aller plus loin et par exemple multiplier les utilités par l'utilité de la configuration U_c (équation (4.28)) laquelle donnera un poids aux utilités déjà calculées, afin de prendre en compte l'alignement de la caméra par rapport au \mathbf{v}_i , donc à la trajectoire que le robot doit exécuter. Comme dans l'approche initiale, nous pouvons aussi anticiper sur l'évolution de cette trajectoire : par exemple en observant plutôt des amers en tournant la caméra vers la gauche, si la trajectoire s'incurve vers la gauche dans les configurations suivantes.

Nous sommes intéressé aussi à minimiser les saccades, sur l'ensemble de la trajectoire, c'est-à-dire à minimiser les modifications des paramètres de la caméra. Il existe plusieurs motivations pour cela : (a) modifier ces paramètres a un coût énergétique, (b) pendant un changement de configuration, des images peuvent toujours être acquises, mais c'est quant même une période aveugle, (c) changer de zoom a des effets secondaires sur tous les traitements qui exploitent le calibrage de la caméra, et enfin, (d) des saccades fréquentes provoqueraient sûrement des ruptures de suivi visuel.

Néanmoins, pour prendre en compte ce problème de minimisation des saccades, nous devons optimiser un critère global pour générer un plan perceptuel optimal sur toute la trajectoire. Jusque là, nous avons considéré optimal, le plan généré en considérant successivement les configurations optimales des paramètres de la caméra pour chaque configuration du robot échantillonnée le long de la trajectoire.

Afin d'optimiser le plan perceptuel au niveau global, considérons que la trajectoire est donnée par une méthode externe de calcul de trajectoires, et échantillonnons la (si elle ne l'est pas déjà) en n_t positions également espacées. Calculons les utilités pour chaque sous-ensemble d'amers visibles au point \mathbf{x}_t , et pour toute position \mathbf{x}_t sur la trajectoire.

A partir de la matrice produite de n_v par n_t valeurs, nous obtenons le meilleur plan perceptuel, au moyen de la programmation dynamique, en minimisant les angles entre deux positions ($\alpha_t - \alpha_{t-1}$) et en maximisant l'utilité des amers, un exemple est donné dans la figure 4.15.

En résumé le plan perceptuel optimale sera un chemin dans le graphe des utilités qui minimise le mouvements saccadiques de la caméra, en considérant les valeurs maximales d'utilités.

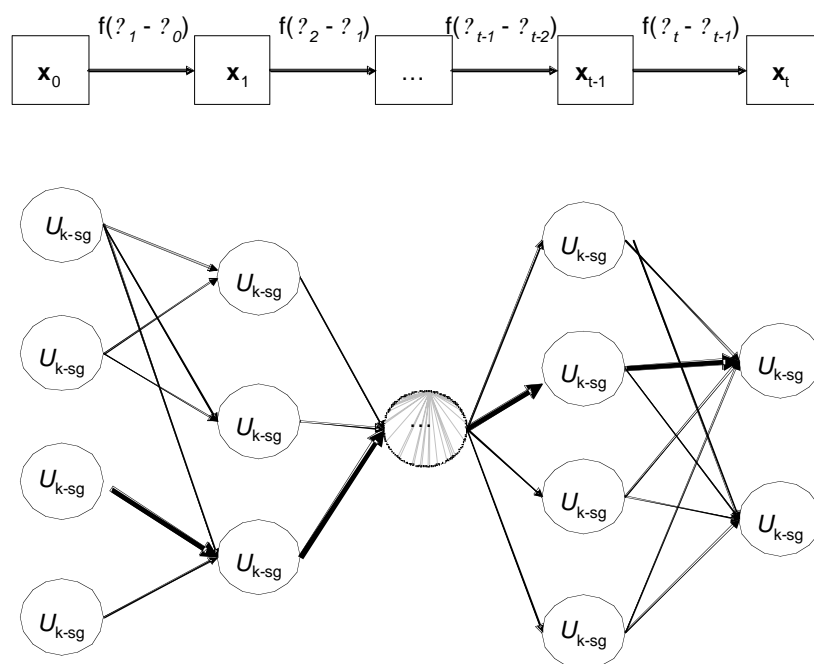
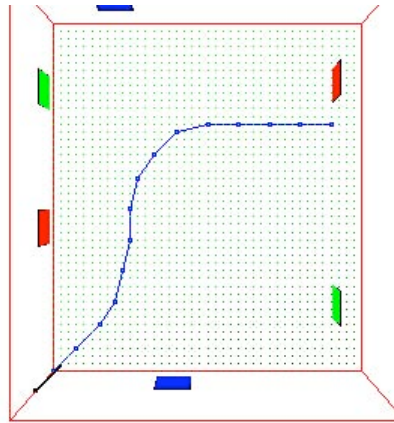
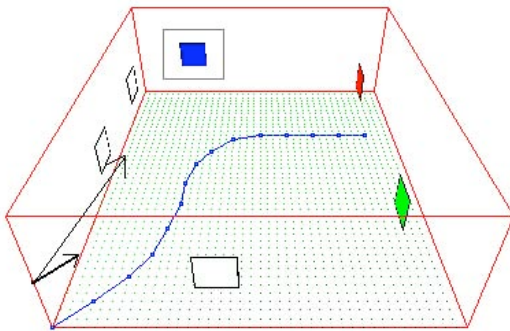


Figure 4.15. Optimisation du plan perceptuel par programmation dynamique ; le graphe représente les combinaisons des cibles le long de la trajectoire. Chaque noeud est l'utilité d'une combinaison de cibles, en un point donné de la trajectoire. Les arêtes joignent les nœuds correspondant à des positions consécutives sur la trajectoire. le plan perceptuel optimal est montré en lignes foncés.

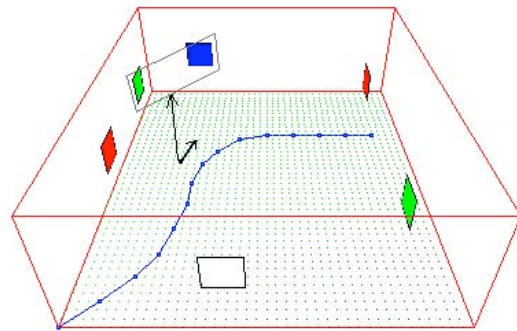
Dans la figure 4.16, nous montrons un exemple des amers visibles et des paramètres optimaux de la caméra pour quelques points sur une trajectoire donnée. Les amers visibles depuis une position donnée \mathbf{x}_t le long de la trajectoire sont montrés en couleur foncée, et le rectangle en trait pointillé montre la ou les cibles choisies, avec l'angle optimale α les vecteurs sortant du point dans la trajectoire montre la direction \mathbf{v}_t du robot par rapport à la trajectoire et le vecteur focal \mathbf{f} .



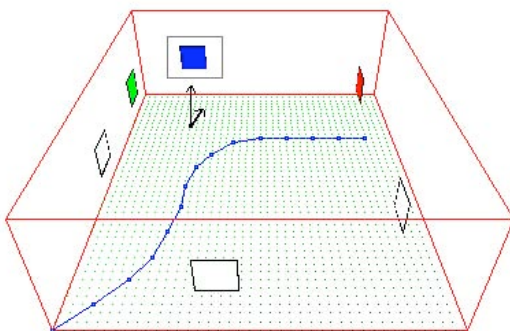
(a)



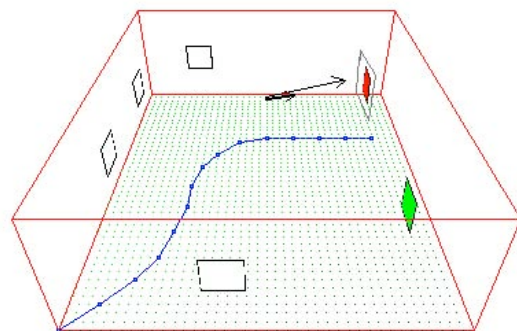
(d)



(c)



(d)



(e)

Figure 4.16. Amers visibles sur une trajectoire et paramètres optimaux de la caméra.

Pour l'instant nous ne prenons pas en compte les occultations des amers, mais l'information des occultations peut être introduite très facilement dans la fonction de visibilité des amers au moyen de plans de visibilité comme ceux utilisés dans la géométrie algorithmique [33].

Finalement signalons une différence importante avec l'approche initiale présentée par V.Ayala-Ramirez ; l'incorporation d'un module embarqué pour la génération en ligne de plans perceptuels le long des trajectoires, est plus facile.

- D'une part, le calcul de l'utilité des amers est directe, et
- D'autre part, dans le cas où la trajectoire est déformée pour éviter un obstacle (stratégie réactive de type *elastic band*), avec la programmation dynamique, il est possible de remplacer quelques noeuds du graphe d'utilité et d'optimiser le plan perceptuel pour la nouvelle section de la trajectoire, sans perdre beaucoup de temps pour le faire.

Cette propriété est très importante ; en effet, en environnement intérieur public, les trajectoires prédéfinies doivent souvent être modifiées en ligne, par exemple, à cause de la présence d'obstacles non définis, ni perçus précédemment, ou par la présence et le déplacement d'humains ou d'autres robots dans le même environnement.

4.3 Enchaînement et permutation entre méthodes de suivi

Dans la section 1.4.2 nous avons mentionné quelques travaux existants qui intègrent plusieurs méthodes de suivi afin d'augmenter la robustesse du système de vision. Ces travaux peuvent être divisés dans deux classes : a) les travaux qui utilisent différentes méthodes de suivi distribuées sur un groupe d'agents, qui coopèrent afin de suivre une cible déterminée, comme dans [107], et b) les travaux, qui utilisent un même agent mais avec différentes méthodes de suivi afin de réaliser la tâche de suivi.

Nos travaux font parti de ce dernier groupe, qui peut être lui-même être subdivisé dans :

- les approches de type **fusion**, avec un module qui combine les résultats de différentes méthodes afin d'augmenter la robustesse et la performance du suivi, comme dans [94], [96] ou [101], et
- les approches de type **contrôle**, avec un contrôleur qui permute entre les différentes méthodes afin de traiter la reprise des erreurs et afin de prendre en compte des conditions variables de l'environnement [30], [105].

Nos travaux se localisent dans ce dernier groupe d'approches.

Dans [105], a été proposée une méthodologie afin d'incorporer plusieurs méthodes de suivi dans un schéma hiérarchique. Cette méthodologie a utilisé une caméra statique et toutes les méthodes de suivi ont été caractérisées pour suivre la même cible, des visages de personnes se déplaçant dans la scène. Une fois que tous les méthodes de suivi ont échoué, soit par une occultation, soit par une disparition de la cible dans le champ de vue de la caméra, le système reste en attente, jusqu'à ce qu'une nouvelle cible apparaisse et soit détectée.

Dans notre cas, d'abord le robot et la caméra sont mobiles ; déjà les mêmes méthodes ne pourront pas être utilisées, ni le même schéma hiérarchique. Ensuite, comme nous l'avons déjà évoqué, les cibles qu'un robot doit suivre sont de nature

différente et sont perçues dans des contextes très variables (en particulier, type du fond de la scène). En plus le robot ne peut pas attendre jusqu'à ce qu'une cible apparaisse à nouveau, dans son champ de vue.

Kruppa et al. dans [69], ont proposé de compléter les travaux de suivi d'objets avec plusieurs caractéristiques dans [101]. Notamment, ils ajoutent une mesure contextuelle, fonction du type d'illumination de la scène. Cependant cette contribution reste limitée ; le suivi se fait avec une combinaison de plusieurs méthodes, mais et la mesure contextuelle sert seulement à sélectionner le modèle colorimétrique à appliquer pour une méthode de suivi de caractéristiques couleurs.

Nos travaux peuvent être vues comme une extension des travaux dans [105], afin d'intégrer dans un modèle hiérarchique, différentes méthodes de suivi pour différentes sortes de cibles et différents contextes. En plus, une fois épuisées toutes les options pour continuer le suivi, une nouvelle cible sera extraite par une autre fonctionnalité visuelle, qualifiée de méthode d'extraction de cible. Une méthode de suivi adaptée pour cette cible et pour les conditions courantes de l'environnement, pourra ensuite être initialisée.

Dans le chapitre 3 nous avons caractérisé chacune des méthodes de suivi, de manière à que le système puisse associer à partir des connaissances de la cible et de leur environnement, la meilleure méthode de suivi, afin d'accroître la probabilité de maintenir le suivi de la cible, le long de la séquence d'images.

Par exemple, quand une méthode de reconnaissance ou un opérateur donne au robot une cible à suivre, le *module de suivi* sera sollicité pour sélectionner la meilleure méthode pour réaliser le suivi de la cible spécifiée. Cela dépendra autant de la cible, comme de l'environnement, et plus précisément des mesures de température autour des trois zones définies précédemment, dans la section 3.4.2.

En même temps, ce module traitera les différentes situations et conditions, pour lesquelles une erreur a pu se produire, ou pour lesquelles une transition pourra être effectuée.

4.3.1 Transition entre méthodes dans la présence des erreurs

Quand une méthode de suivi donnée détecte une erreur, deux stratégies peuvent être mises en oeuvre :

- a) le système sélectionnera une nouvelle méthode pour essayer de maintenir la tâche de suivi de la même cible, ou
- b) si la nouvelle méthode n'a pas pu reprendre le suivi de la cible (échec immédiat), ou s'il n'existe aucune autre méthode pour réaliser la tâche avec les conditions courantes, le module de suivi reportera l'erreur au système décisionnel et demandera au module de reconnaissance d'objets la cible suivante à suivre dans le plan perceptuel.

La figure 4.17 montre le suivi d'une affiche plane dans un fond encombré. Dans ces conditions la méthode sélectionnée est le suivi par différences de motifs. Le suivi se déroule bien entre les images (a) et (d), cependant, dans l'image (e) la méthode commence à dériver et en pratique, il détecte qu'il a perdu la cible dans l'image

suivante. L'image dans (f), montre le résultat, si le suivi avait été poursuivi avec la même méthode.

En se fondant sur les caractéristiques autant de la cible que de l'environnement, le module sélectionnera une nouvelle méthode, dans ce cas la méthode choisie pour essayer de reprendre l'erreur, est le suivi d'un ensemble de points, également adaptée pour des cibles texturées et qui n'a pas besoin de connaissance a priori, excepté un modèle de cible. Ce modèle est fourni par l'ensemble de points d'intérêt extraits de la dernière représentation valide de la cible, à savoir l'image (d). Dans la figure (g), nous présentons la même image que dans la figure (f), sauf que c'est le résultat du suivi d'un ensemble de points qui est présenté. Dans ce cas, la transition entre les méthodes a été un succès, et l'image (h) montre la continuation de la tâche de suivi pour la même cible, toujours avec le suivi par un ensemble de points.

Pour plus de précision, dans les images (a) à (f), où la méthode de différences de motifs a été utilisée, le contour en trait continu (vert) représente la position de la cible trouvée après l'approximation du vecteur de mouvement, et le contour en trait pointillé montre la prédiction, laquelle n'est que l'ajustement dans l'image précédente. Dans les images (g) et (h), où la méthode utilisée, est celle du suivi d'un ensemble de points, le contour en trait pointillé représente la zone de recherche et le contour en trait continu, le meilleur appariement trouvé pour la cible dans cette zone.

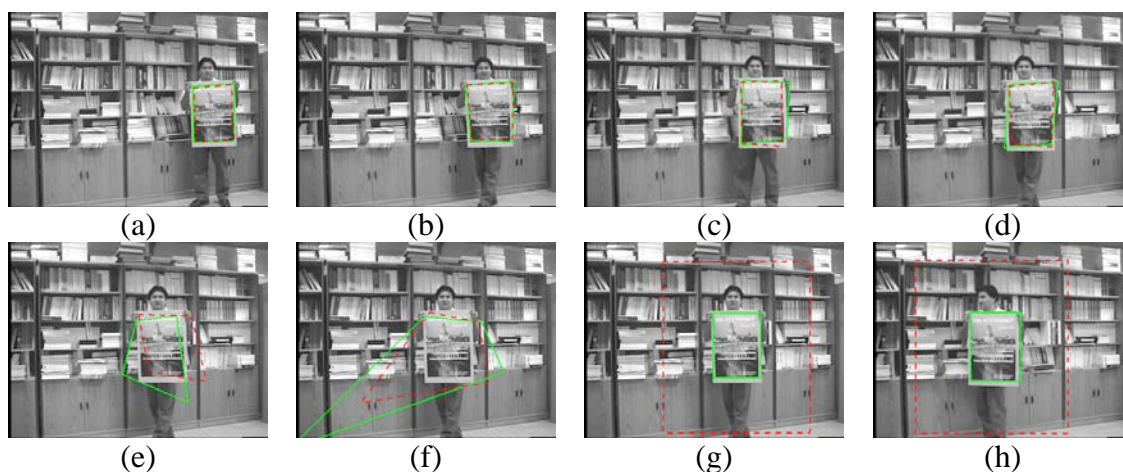


Figure 4.17. Transition après la détection d'erreur d'une méthode de suivi. De (a) a (f) suivi avec différences de motifs ; en (g) et (h) suivi d'un ensemble de points.

Comme c'est la première méthode qui est supposée être optimale pour le suivi d'une cible spécifique dans les conditions courantes, si pendant un certain temps t_e , la deuxième méthode choisie, continue de suivre la cible, il sera possible de retourner à la première méthode. Pour cela pendant quelques itérations les méthodes seront effectuées en concurrence.

Par contre, il n'y aura pas de retour à la méthode considérée initialement comme optimale, dans le cas où les conditions de l'environnement ont changé (typiquement pour une caméra mobile, le fond de la scène est devenu uniforme ou encombré).

La figure 4.18, propose un schéma de transitions entre méthodes de suivi. Ce schéma a été défini en fonction des résultats de l'évaluation et de la comparaison des méthodes du chapitre 3.

Cependant, ce schéma ne garantit pas que notre module de suivi ne perdra pas la cible ; par exemple, avec des changements d'illumination, comme ceux montrés dans la

figure 3.11, la texture de la cible peut disparaître, et donc il sera presque impossible de le suivre avec les méthodes de suivi de régions, que ce soit par différences de motifs, ou par un ensemble de points. Donc une solution pour ce contexte peut être le suivi avec lignes, si les contours sont suffisamment forts ou avec les modèles de contours actifs, si l'environnement n'est pas texturé.

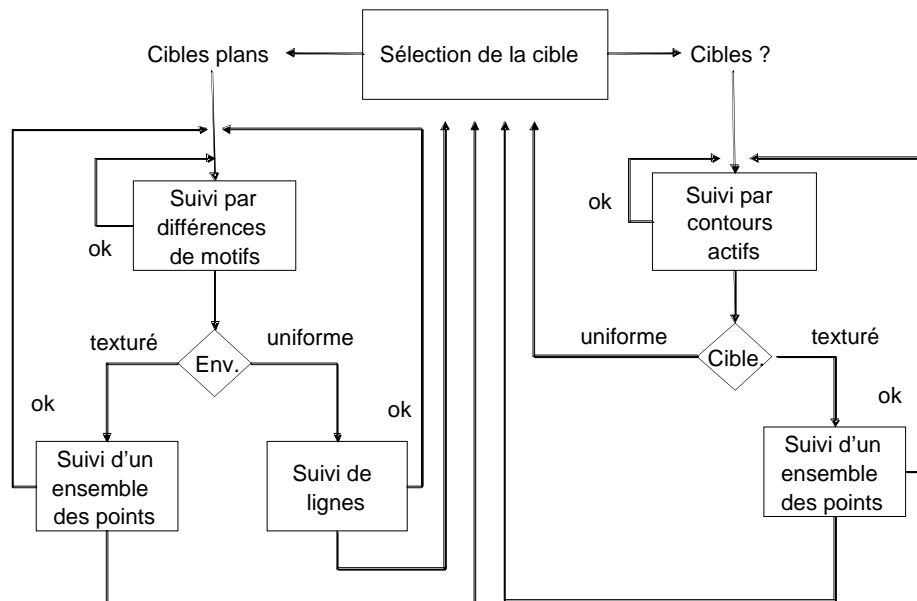


Figure 4.18. Sélection des méthodes de suivi en fonction des caractéristiques de la cible et des erreurs dans le processus de suivi.

4.3.2 Changements des cibles

Un changement de cibles, qui peut s'accompagner possiblement d'une commutation entre deux méthodes de suivi, si les caractéristiques des cibles à suivre sont différentes, sera effectué en plusieurs cas. Citons :

- dans le plan perceptuel on arrive à un point x_t de la trajectoire, dans lequel une transition est prévue ; il faut, soit augmenter le nombre de cibles à suivre, soit simplement changer la cible courante par la suivante
- toutes les méthodes disponibles pour le suivi d'une cible spécifique ont échoué, du fait soit d'occultations de la cible, soit de problèmes avec les conditions d'illumination, soit d'un changement sur la trajectoire à cause d'obstacles ...ou autres. Une transition sera alors décidée en ligne, avec éventuellement un changement dans le plan
- Une transition peut être demandée explicitement par un autre module pour une tâche spécifique du robot (par exemple, *Suis tel obstacle mobile* pour initier une navigation en convoi)
-

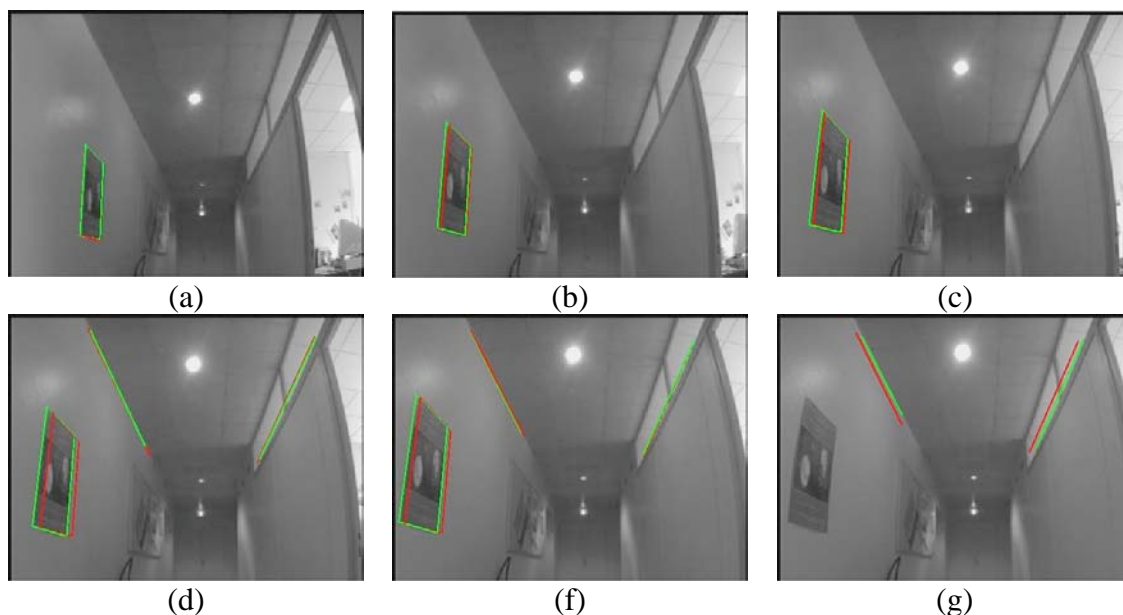


Figure 4.19. Transition entre méthodes de suivi pour des cibles différentes.

Dans le premier cas, la transition pourra se faire de manière graduelle, c'est-à-dire que pendant quelques itérations les méthodes assignées à chaque cible s'exécuteront en concurrence, comme est montré dans la figure 4.19. Dans cette figure est présenté, le suivi d'un amer plan dans un couloir. Cette cible est suivie, toute seule entre les images de (a) à (c), et à un moment donné la consigne est de changer de cibles, dans ce cas, de suivre les lignes droites arêtes entre mur et plafond du couloir. Afin d'éviter les changements brusques de méthodes et / ou de cibles, pendant quelques itérations (typiquement 3 images) toutes les cibles sont suivies, ce qui correspond aux images (d) et (f). Finalement, si le suivi sur les nouvelles cibles est correct, le module reste seulement sur leurs suivis (image (g)).

Dans le deuxième cas, il faudra simplement regarder le plan perceptuel et vérifier et modifier les paramètres de la caméra afin de mettre la cible suivante dans le de vue de la caméra. Il est possible alors qu'aucune cible ne soit suivie pendant quelques instants, exactement pendant l'ajustement des paramètres de la caméra, ou pendant qu'un module d'extraction, reconnaissance ou localisation recherche la cible suivante indiquée dans le plan perceptuel.

La figure 4.20, montre un schéma dans lequel n cibles sont suivies en concurrence, par n méthodes de suivi, qui peuvent être les mêmes ou non, selon les caractéristiques de la cible et de son environnement. Dans notre cas n est limité à 3 cibles comme maximum, pour ne pas affecter le temps de réponse du système.

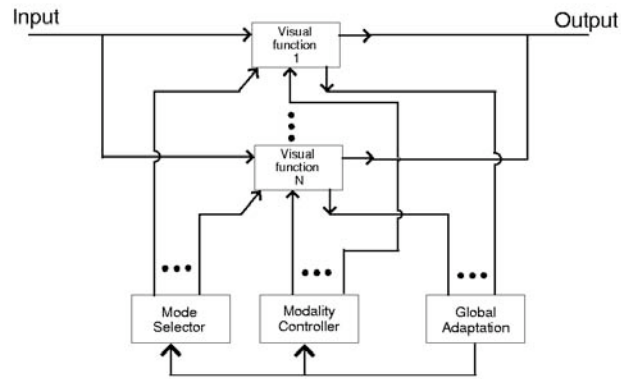


Figure 4.20. Diagramme des fonctionnalités visuelles.

4.3.3 Autres modules : contrôleur du module de suivi, observateurs

Finalement, pour réaliser ces fonctions de reprise d'erreurs, ou de contrôle d'exécution d'un plan perceptuel afin de faire des commutations, il sera nécessaire d'introduire un ou des modules spécifiques, différents du module de suivi abondamment décrit jusqu'alors. Ce contrôleur choisira des nouvelles cibles en fonction d'un plan perceptuel en cours d'exécution, ou enverra des requêtes afin d'arrêter un suivi en cours d'une cible donnée afin de commuter avec une autre cible, ou traitera les bilans d'erreur des méthodes en cours d'exécution ... Ce module est le module de contrôle de suivi évoqué dans la figure 1.7.

Il va gérer les transitions entre cibles, donc possiblement entre méthodes de suivi.

Comme dans les travaux de R.Pissard-Gibollet et P.Rives, les transitions pourront être déclenchées par la détection d'évènements visuels, comme la détection de lignes verticales, de zones texturés, ou simplement de variations de luminance dans une fenêtre prédéfinie. Ces évènements visuels, pourront être détectés, soit par détection au niveau pixel (*pixel-level*), comme dans [87], ou au niveau région (*blob-level*), comme dans [114].

Cette recherche d'évènements se distingue de l'extraction d'amers ou de primitives, car elle se fait uniquement localement. Les zones où est recherché un évènement visuel, pourront être liées au suivi d'une cible spécifique comme par exemple : à une certaine distance d'une ligne droite qui est suivie, comme il est montré dans la figure 4.21. Dans cette figure l'évènement visuel cherché est une ligne verticale, laquelle correspond à une arête mur / porte.

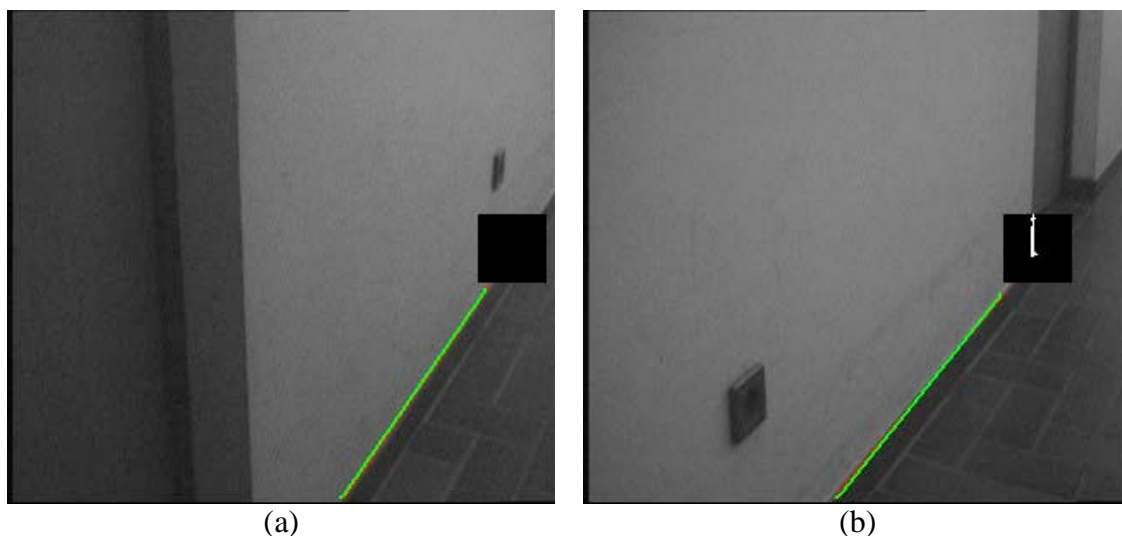


Figure 4.21. Détection des événements visuels, dans une zone liée au suivi de ligne.

4.4 Conclusions

Dans ce chapitre, nous avons décrit les approches développées initialement dans le groupe, afin d'obtenir une mesure sur la visibilité des cibles depuis une position donnée (travaux de JB.Hayet), et afin d'obtenir un plan perceptuel, pour la navigation métrique (travaux de Ayala-Ramirez). C'est à partir de l'analyse de ces travaux, que nous avons proposé une manière alternative, autant pour le calcul de la visibilité que pour la génération de plans perceptuels.

La nouvelle mesure de la visibilité a été élaborée de manière à considérer les dimensions et orientations des cibles, ainsi que la distance à celles-ci, mais sans prendre en compte les paramètres de la caméra. Ces paramètres ont été introduits dans le calcul de l'utilité d'un amer quand le robot est dans une configuration donnée ; cette utilité, dépend donc de la caméra utilisée, et les valeurs d'utilité pour tous les amers, seront exploitées afin de trouver la configuration optimale des paramètres de la caméra, pour un ensemble d'amers visibles.

Par la suite une méthode basée sur la programmation dynamique a été proposée, afin de générer le plan perceptuel en traitant la minimisation de saccades sur toute la trajectoire du robot.

Finalement, une méthodologie pour la collaboration et la transition entre les méthodes de suivi a été proposée. Elle dépend, d'une part, de la génération du plan perceptuel, et d'autre part, de la caractérisation proposée dans le chapitre précédente, pour les méthodes de suivi en fonction de la nature des cibles et des environnements. Cette méthodologie prend en compte aussi, la détection d'erreurs spécifiques pour chacune des méthodes de suivi, ainsi que les différents contextes dans lequel le robot peut évoluer.

Chapitre 5

Intégration et expérimentation

5.1 Introduction

Comme nous l'avons déjà remarqué, notamment au chapitre 3, les méthodes de suivi ont besoin d'un algorithme de reconnaissance d'objets afin que, d'une part, on puisse sélectionner la méthode à utiliser, en fonction des caractéristiques extraites de la cible, et d'autre part, on puisse initialiser le module de suivi avec la première position de la cible. De manière duale, comme mentionné au chapitre 1, les méthodes de suivi peuvent limiter la complexité des méthodes de reconnaissance, en fournissant des prédictions sur la position des objets à reconnaître, ce qui permet d'éviter la détection en continue des cibles.

Dans beaucoup d'approches, la méthode de reconnaissance d'objets sert aussi de méthode de suivi. Cependant les approches utilisées en ce cas sont, la plupart du temps, très rudimentaires et/ou limitées à suivre des caractéristiques spécifiques ; citons par exemple la détection d'obstacles sur chaque image d'une séquence, sans tenir compte du résultat de la détection sur l'image précédente.

Comme mentionné dans le chapitre 1, l'architecture LAAS, comporte trois niveaux (figure 1.5). Parmi eux, c'est le niveau fonctionnel qui nous intéresse, parce que toutes les fonctionnalités visuelles y sont intégrées dans des modules, avec les autres fonctions sensori-motrices du robot (figure 5.1). Dans cette couche fonctionnelle, il est possible de créer plusieurs modules indépendants, qui communiquent par messages (interactions requêtes/répliques/bilan) ou par le biais de zones mémoire communes, ou variables partagées appelées posters.

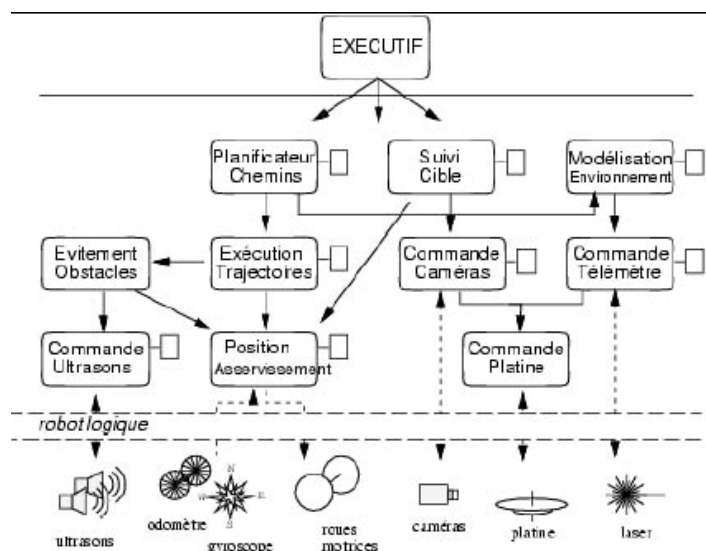


Figure 5.1. Exemples de modules à l'intérieur de la couche fonctionnelle dans l'architecture LAAS.

En fait, nous nous sommes intéressé uniquement aux modules relatifs à la vision ; même si ces modules doivent coopérer avec d'autres modules pilotant les actionneurs (asservissement visuel, évitement d'obstacles...) ou traitant les données acquises par d'autres capteurs (localisation sur segment laser, fusion vision/LASER ...), nos travaux en resteront là. Signalons toutefois que la commande des paramètres de la caméra (ou de la platine qui la porte) a également été traitée.

Plus précisément, nous souhaitons intégrer le module de suivi visuel que nous avons développé, avec différents modules de reconnaissance de caractéristiques ; la nature de cette reconnaissance dépendra du contexte pour lequel ces modules seront utilisés (figure 1.7).

Dans les sections suivantes nous allons détailler notre approche pour la commande des paramètres de la caméra, pour ensuite décrire, parmi celles définies au chapitre 1, l'intégration des différentes méthodes d'extraction des caractéristiques et des contextes dans lesquels elles seront exploitées. Un utilitaire de simulation d'un environnement intérieur sera aussi décrit; il a été exploité, autant pour tester les algorithmes de suivi, que pour tester leurs réactions lorsque sont modifiés en ligne, les paramètres de la caméra active.

5.2 Commande des paramètres de la camera

Comme dit dans le chapitre 1, la commande des paramètres de la caméra répond à un besoin primordial de la vision active.

Entre toutes les possibilités pour effectuer la commande des modalités site, azimut et zoom de la caméra, nous avons choisi de le traiter à partir de la logique floue. L'implémentation est plus facile ; pour réaliser par exemple un contrôleur PID, on a besoin d'un modèle le plus précis possible du système à contrôler, généralement représenté par un ensemble d'équations différentielles ; la caméra utilisée, une caméra de vidéo-surveillance bas-coût (Sony EVI-D31), n'est pas adaptée pour des approches aussi formelles, puisque les actionneurs sont des moteurs pas à pas et qu'il n'est pas réaliste de faire une commande en vitesse très « propre ».

A la différence des autres contrôleurs, un contrôleur flou ne demande pas un modèle précis du système à commander ; les consignes sur les moteurs sont déterminées à partir de décisions prises sans avoir recours à un modèle analytique.

Les algorithmes de commande, se basent sur des règles linguistiques de la forme, « *si 'X' alors 'Y'* ». Nous pensons que ce type de formalisme est plus adapté dans le contexte du développement d'un système autonome; dans le futur, le robot pourra modifier ces règles ou pourquoi pas, en apprendre de nouvelles, sans avoir besoin de l'intervention d'un opérateur, mais seulement par des techniques d'apprentissage fondées sur les propres expériences du système.

La logique floue, concept introduit en 1965 par Lotfi Zadeh, propose une approche des problèmes beaucoup plus pragmatique que mathématique, ce qui, dans de nombreux cas, n'est pas du tout accepté par des esprits cartésiens ou trop déterministes. Pourtant, les solutions proposées aux problèmes par la logique floue, ont des résultats similaires aux autres approchés [92]. Comme tout nouveau concept, la logique floue nécessite un effort de compréhension ou plutôt une disponibilité d'esprit. En plus, pour certains problèmes où les mathématiques ont une difficulté ou impossibilité pour la modélisation, la logique floue apporte une très bonne efficacité.

La logique floue est très proche du processus de la pensée humaine quotidienne. Elle met en oeuvre un jeu de règles comme, implicitement, nous en utilisons chaque jour. Pourtant, les règles nécessaires pour la commande floue, peuvent être exprimées en utilisant le langage de tous les jours et la connaissance intuitive d'un opérateur humain.

Nous avons divisé le contrôleur en deux composantes : a) le contrôle de l'angle de champ visuel (*zoom*), et b) le contrôle des angles azimut et site.

5.2.1 Angle du champ visuel - Zoom

Afin de réaliser la commande pour l'angle de champ de vue de la caméra, nous allons tout d'abord décrire les différentes composantes de notre système sans entrer beaucoup dans les détails sur les aspects de la logique floue, qui peuvent être consultés dans [92].

Un contrôleur flou, est composé de plusieurs composantes, parmi lesquelles on peut nommer : a) la quantification floue des entrées et sorties, b) la base de règles, et c) la combinaison de règles.

La première étape consiste donc à modéliser chacune des entrées et sorties du système par des courbes donnant les degrés d'appartenance des variables à différents états prédéfinis qui caractérisent un mode de fonctionnement du système (*fuzzyfication* et *défuzzyfication*). Une fois que la base de règles est établie, on peut définir la manière avec laquelle les règles seront combinées. Par la suite nous donnons des exemples de ces règles : « *Si la taille de la cible es plus petite de la taille désire augmenter réduire le champ de vue* », ou « *Si la distance du centre la cible au centre de l'image est positive et la vitesse de mouvement est négative, donc appliquer une vitesse positive à \square* »

La commande de l'angle \square consiste à adapter l'angle de champ de vue, comme défini dans le chapitre 4 , pour qu'un objet qui doit être perçu dans l'image, ait une taille apparente donnée. Cette taille est déterminée par la diagonale de la boîte englobante de

l'image de cet objet. Nous utilisons comme variable d'entrée la différence ou erreur e_d entre la taille courante et la taille souhaitée pour une boîte englobante dans l'image. Egalement, nous exploitons comme variable d'entrée, la variation de cette différence par rapport au temps, \dot{e}_d .

La sortie du système sera la vitesse v_{\square} , qui devra être appliquée au moteur de la caméra afin d'arriver à l'angle de champ de vue souhaité.

Les deux variables d'entrée, ainsi que la variable de sortie, seront quantifiées dans trois états appelés, *descripteurs linguistiques*, décrits dans la table 5.1.

	Entré		Sortie
	e_d	\dot{e}_d	v_{\square}
Descripteurs linguistiques	Négative	Négative	Négative
	Zéro	Zéro	Zéro
	Positive	Positive	Positive

Table 5.1. Quantification de variables d'entrée et sortie

La base de règles est générée à partir de toutes les combinaisons des valeurs des descripteurs linguistiques des variables d'entrée, et chacune de ces règles définira comme sortie, une valeur d'un descripteur linguistique de la variable de sortie. Ces valeurs seront données initialement par un opérateur, mais on peut envisager que cette base soit modifiable par le robot lui-même, par une méthode d'apprentissage exploitant les résultats obtenus sur la commande de cet angle du champ de vue. D'autres descripteurs, et donc d'autres règles combinant les variables d'entrée, pourraient se révéler nécessaires pour améliorer le comportement du zoom.

		e_d		
		Négative	Zéro	Positive
\dot{e}_d	Négative	P	Z	P
	Zéro	P	Z	N
	Positive	N	Z	N

Table 5.2. Base de règles pour la commande du zoom, les lettres N, Z, P, représentent les descripteurs linguistiques de la variable de sortie.

Afin d'obtenir la valeur de sortie finale ou totale, il faut combiner les valeurs des règles actives, ce qui est fait par la méthode de la Racine de la Somme des Carrés, connue par son mnémonique en anglais *RSS (Root-Sum-Square)*, Cette méthode combine les effets de toutes les règles applicables, met à l'échelle les fonctions et leur magnitudes, pour ensuite calculer le centre de la surface ainsi déterminé. Cette méthode assure des valeurs de sortie dans tout l'intervalle continu définie par la quantification.

5.2.2 Angles azimut et site

De manière similaire à la commande de l'angle de champ de vue de la caméra, la commande pour ajuster les angles azimut et site, seront faits en vitesse, v_{α} et v_{β} respectivement. Cet ajustement est fait pour imposer une position donnée à la projection d'un objet dans l'image, typiquement le centre de l'image, afin d'optimiser la probabilité de conserver une cible dans l'image. Donc les variables d'entrée seront d'une part, l'erreur ou différence entre la position souhaitée dans l'image, et la position calculée du centre de la boîte englobante de l'image de l'objet, et d'autre part, comme dans le cas de l'angle θ la vitesse de changement de cette erreur.

Le processus d'ajustement des paramètres site et azimut, peut être divisé en deux parties découplées, l'une pour annuler les composantes horizontales de l'erreur et l'autre pour annuler les composantes verticales de cette même erreur, afin d'ajuster les angles d'azimut α et de site β respectivement. Afin d'avoir un ajustement plus fin de ces angles par rapport à l'ajustement du champ de vue θ nous allons quantifier, autant les variables d'entrée que celles de sortie, dans les cinq descripteurs linguistiques suivantes : *Négative, Peu Négative, Zéro, Peu Positive, et Positive*.

Comme dans le cas précédent, la base de règles sera générée par la combinaison de tous les descripteurs linguistiques des variables d'entrée et la valeur de sortie pour chaque règle sera initialement donnée par un opérateur.

La combinaison des règles pour obtenir la valeur finale de la variable de sortie sera faite, comme dans le cas précédent par RSS ; cette combinaison survient dans notre cas, lorsque la caméra est contrôlée pour conserver plusieurs objets dans l'image. Comme il n'est pas possible en ce cas de centrer les deux en même temps, une valeur moyenne de centrage est déterminé par combinaison ; en pratique, le centrage se fera sur le milieu de la droite reliant les deux boîtes englobantes dans l'image.

Un point important à considérer pour le calcul des valeurs de sortie, et plus précisément pour les valeurs numériques des descripteurs linguistiques, est que la vitesse maximale des variations des angles α et β , doit permettre aux méthodes de suivi d'objets de suivre leur cible, dans les conditions satisfaisantes. C'est-à-dire, le mouvement apparent des cibles dans les images, rajouté par la commande de la caméra active, ne doit pas dépasser un seuil donnée, afin de ne pas perturber les méthodes de suivi, qui rappelons-le, se fondent sur des critères locaux, donc supposent de faibles mouvements apparents. Remarquons que :

- en principe, le centrage des cibles tend à annuler ce mouvement apparent.
- cette vitesse peut être ajustée au moment de la quantification des valeurs de sortie, enfin, il existe un couplage entre commandes de l'orientation et commande du champ de vue, puisque la modification du champ de vue crée lui aussi un mouvement apparent radial dans les images.

Notons que, la sortie de l'ajustement du champ de vue de la caméra sera utilisée pour limiter les valeurs de quantification des descripteurs linguistiques pour les variables de sortie.

Dans la figure 5.2 nous montrons un schéma simplifié du fonctionnement du contrôleur.

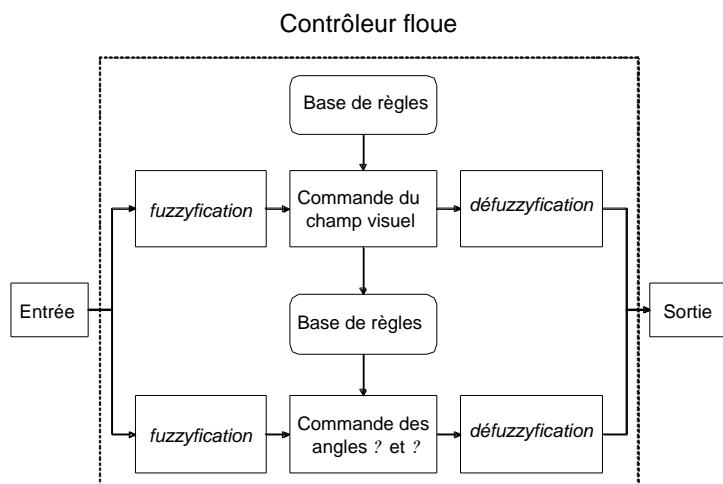


Figure 5.2. Schéma simplifié du contrôleur.

5.2.3 Résultats du contrôle d'une caméra active

Il serait intéressant de comparer cette approche de la commande floue, avec des méthodes plus classiques de l'asservissement visuel, en particulier inspirées du concept de la commande référencée tâche (F.Chaumette). Nous n'avons pas eu le temps de mener cette comparaison.

Nous avons intensivement validé le contrôleur flou mis en œuvre pour asservir l'orientation et le champ de vue d'une caméra active, soit depuis une caméra embarquée sur un robot, soit depuis une caméra fixe, donc dans un contexte vidéo-surveillance. Nous montrons ci-dessous des résultats en simulation ; d'autres résultats apparaîtront dans les expérimentations réelles.

5.3 Simulation (environnement virtuel d'intérieur)

Afin de tester autant le contrôleur flou de notre caméra active, que les méthodes de suivi nous avons fait un simulateur pour des environnements d'intérieur. Dans la figure 5.3, nous montrons les résultats d'une simulation : un robot portant une caméra active, décrit une trajectoire circulaire dans un espace ouvert, et exploite sur cette trajectoire, les amers de type poster rajoutés sur les murs, afin de se localiser. Le simulateur est uniquement dédié aux traitements visuels et à la commande de la caméra.

La figure montre deux vues aériennes (vue de dessus et vue perspective) : y apparaissent l'environnement (en gris clair pour les murs, foncé quadrillé de rouge pour le sol), les posters projetés sur les murs dans des positions définies par l'utilisateur ; le robot est représenté par le cylindre noir ; le cône vert symbolise les paramètres de la caméra montée sur le robot : l'orientation et le champ de vue.

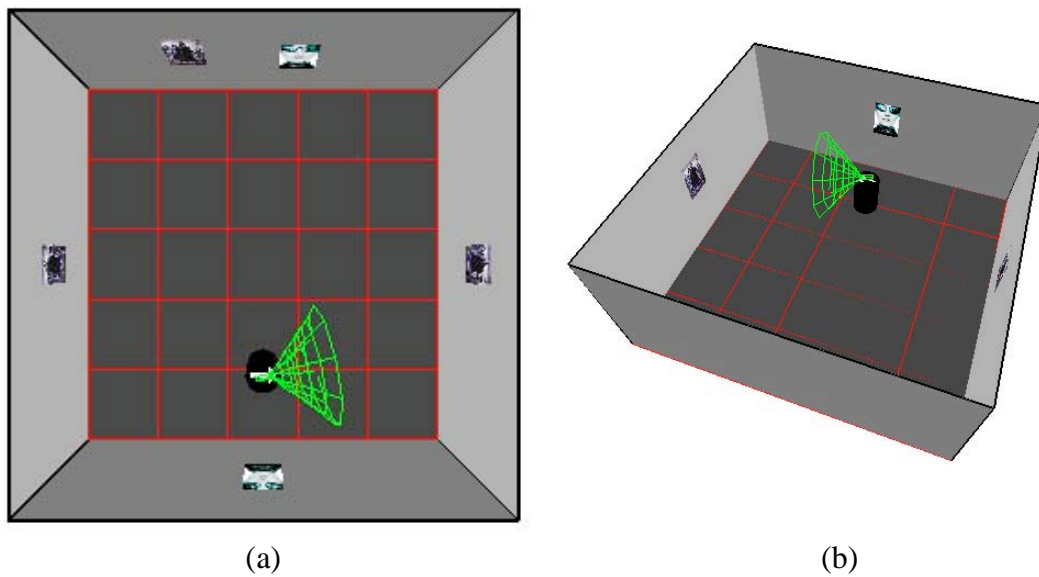


Figure 5.3. Deux vues de dessus produites par le simulateur.

Dans cet environnement simulé, il s'agit de valider nos modules dans un contexte de navigation métrique. Le robot doit décrire une trajectoire circulaire ou elliptique dans une salle. Le long de la trajectoire le robot trouvera des affiches qui lui serviront pour actualiser sa localisation par rapport à un repère du monde. Le but principal est de suivre les affiches le plus longtemps possible, en contrôlant l'orientation et le champ de vue de la caméra. Une fois perdues, à cause de la position et/ou du mouvement du robot, le système va sélectionner la configuration suivante de la caméra dans le plan perceptuel, afin d'observer les affiches ou cibles suivantes ; dans ce plan, le robot traite de l'enchaînement entre phases de suivi, effectué en détectant deux amers dans certaines images de la séquence.

Pour l'instant la reconnaissance des objets ne se fait pas automatiquement ; l'initialisation des phases de suivi est faite par un opérateur par des clics souris. Mais une fois finie cette initialisation, le processus de suivi, comme l'ajustement des paramètres de la caméra active, sont effectués automatiquement, par les méthodes décrites précédemment. La détection automatique des cibles sera incorporée dans l'implémentation réelle sur le système embarqué.

La figure 5.4, montre l'ajustement des paramètres de la caméra. Les images de (a) à (d) montrent des vues aériennes de la scène virtuelle. Rappelons que l'axe du cône représente l'axe optique f et l'ouverture du cône représente l'ouverture du champ visuel. Dans les images (e) à (h) sont montrées les images de la scène acquises dans la position courante du robot avec les paramètres estimés de la caméra ainsi que le résultat du suivi de la cible, dans ce cas un affiche dans le mur.

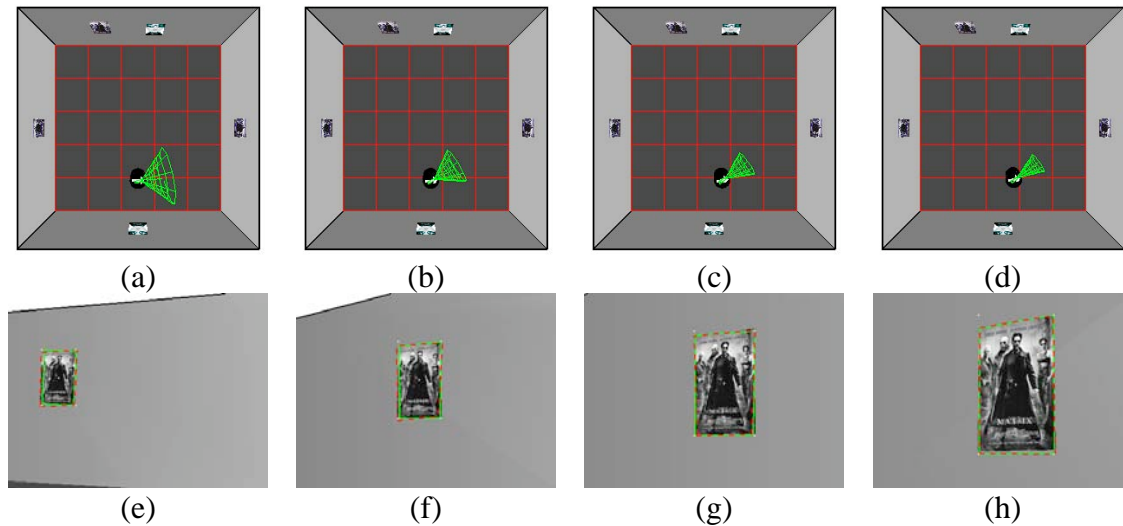


Figure 5.4. Ajustement des paramètres de la caméra. En haut, position du robot et champ de vue de la caméra dans la scène. En bas, images virtuelles générées par le simulateur et données en entrée, au module de suivi des amers.

Le suivi est fait par la méthode de différences de motifs. Le contour en lignes pointillées (rouge) représente l'hypothèse initiale et le contour en lignes continues (vert) représente l'ajustement fait par la différence de motifs.

La simulation permet de faire varier la fréquence de génération des images ; de ce fait, les considérations sur la vitesse autant du robot comme de la caméra sont prises en compte et les images de synthèse sont générées en simulant une fréquence de 20hz.

La figure 5.5 montre le suivi de différentes cibles pendant que le robot suit sa trajectoire. Dans les vues (a) à (c), le robot suit la cible depuis les images acquises présentées de (e) à (f) ; il ajuste les paramètres de la caméra pour avoir une image de la cible de taille presque uniforme. Ces paramètres sont représentés par le cône au-dessus du robot, ce qui nous permet d'apprécier les variations produites sur les paramètres.

Quelques instants après, dans la vue (d), le robot ne peut pas continuer à suivre la même cible du fait des configurations limite de la caméra active et de sa position dans l'espace ; l'image (h) est acquise pendant le mouvement du robot et de la caméra afin de retrouver un amer sur le mur suivant. Le système sélectionne de nouveaux paramètres pour la caméra, à partir du perceptuel, pour arriver à la configuration montrée dans (i), depuis laquelle l'image de synthèse montrée en (m) peut être acquise.

A ce moment la localisation des deux nouvelles cibles pour le suivi est faite, dans ce cas les deux affiches présentes sur le mur dans la vue (i). Le suivi peut reprendre sur ces deux cibles simultanément, avec un ajustement des paramètres de la caméra sur la boîte englobante de ces deux cibles (images (j) et (n)).

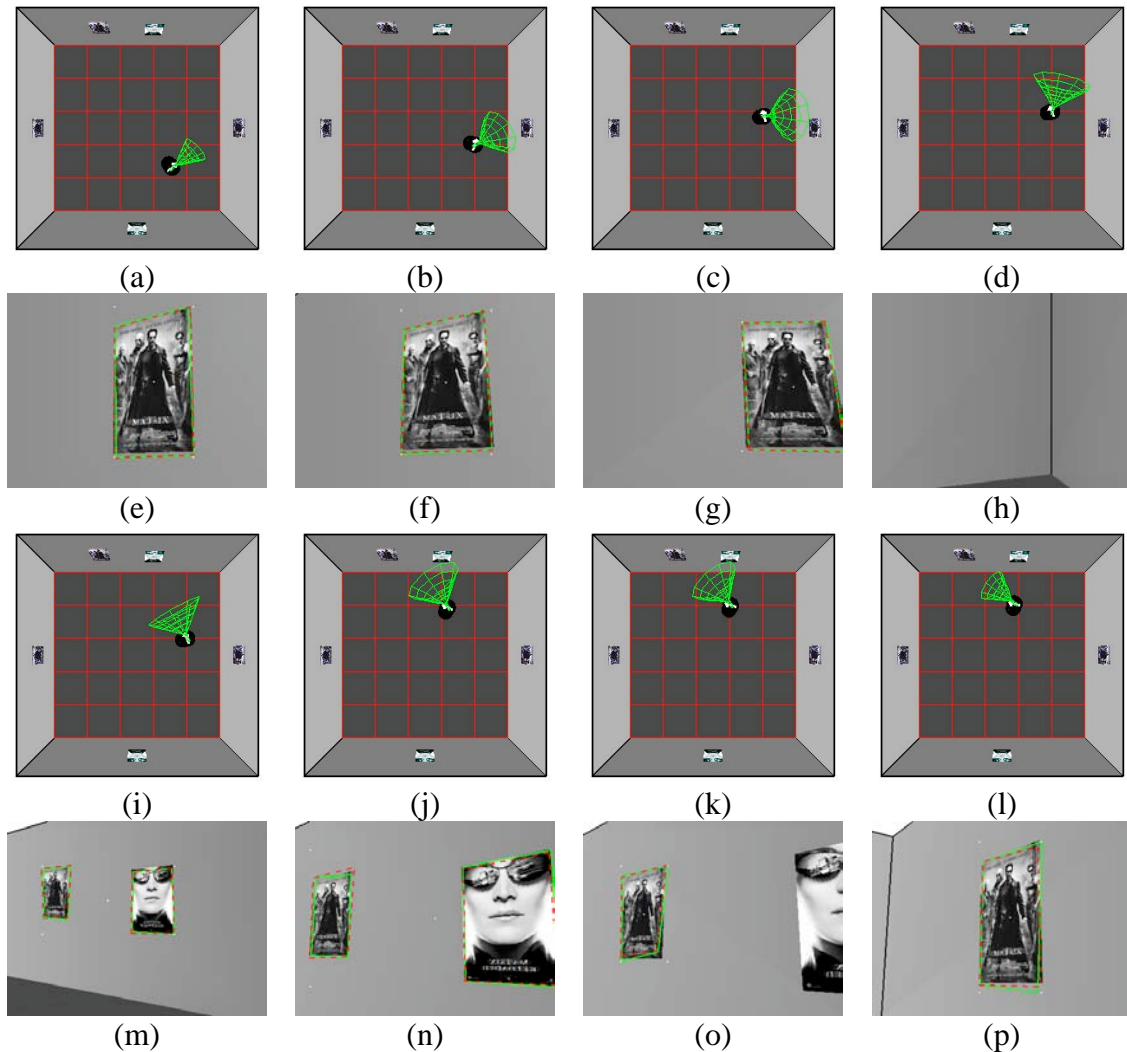


Figure 5.5. Suivi et changement de cibles pendant l'exécution d'un plan perceptuel. Simultanément, ajustement des paramètres de la caméra active.

A un moment donné, dans la situation illustrée dans la vue (k), un des deux amers sort de l'image acquise montrée en (o) ; le robot ne peut pas continuer à suivre les deux cibles, et donc il va continuer à suivre la plus lointaine de ces cibles, ce qui lui est indiqué dans le plan perceptuel. Il va donc ajuster maintenant les paramètres de la caméra pour ces conditions, comme cela apparaît dans la situation montrée dans la vue (l) depuis laquelle l'image (p) est acquise.

5.4 Intégration et résultats expérimentaux

Différentes expérimentations ont été réalisées et vont être décrites dans la suite :

- une dans un contexte intérieur : pour la navigation topologique basée sur le suivi des primitives. Notons que dans ce cas, le robot est commandé par l'opérateur ; il conviendrait d'intégrer nos traitements avec des modules d'asservissement visuel pour réaliser cette dernière expérimentation de manière complète.

- deux dans un contexte extérieur a) pour la navigation d'une machine qui doit suivre une route, et b) pour la détection et le suivi d'obstacles pour des véhicules « intelligents » en milieu routier.

5.4.1 Intérieur – navigation par suivi d'amers

Cette première expérimentation est significative du premier contexte pour la navigation en milieu intérieur. Il s'agit de réaliser dans des conditions réelles, le scénario joué avec le simulateur dans la section précédente. Dans un espace ouvert, des amers sont disposés sur les murs ; les amers sont appris par le robot, grâce aux travaux de JB.Hayet, et rajoutés à une carte stochastique durant une phase d'apprentissage hors ligne.

Une trajectoire fermée est générée : le robot doit réaliser des tours dans cet espace ouvert. Un plan perceptuel est généré également pour sélectionner en tout point le ou les meilleurs amers à exploiter pour la localisation, ainsi que les paramètres optimaux de la caméra active pour réaliser le suivi visuel de ces amers et pour enchaîner les phases de suivi.

Le robot exécute à la fois les déplacements le long de cette trajectoire et le plan perceptuel. Plusieurs modules doivent interagir pour ce faire :

- Contrôle d'exécution du plan perceptuel : c'est le chef d'orchestre, qui lance des requêtes sur les autres modules.
- Module de suivi visuel
- Module de contrôle de la caméra active
- Module de reconnaissance et localisation des amers (module LOCPOST pour *Localisation Poster*, module développé par JB.Hayet) ; ce module doit interagir avec le module de suivi visuel (a) pour prédire la position de l'amer devant servir à localiser le robot, sur la base du suivi visuel et (b) pour utiliser des paramètres intrinsèques de la caméra correspondant à sa configuration courante.
- Module de gestion de la localisation du robot (module POM pour *Position Manager*) qui fusionne de manière probabiliste, les localisations issues des différentes modalités embarquées (laser, odométrie, vision pour le robot DILIGENT)

Cette expérimentation a été intégrée partiellement sur le robot DILIGENT. Nous avons pu réaliser dans un monde réel, une démonstration similaire à ce que nous avons fait sur le simulateur :

- à l'initialisation, d'un amer à suivre en fonction de la position courante du robot, pointage de la caméra active sur la position prédite de cet amer,
- recherche de l'amer à suivre afin d'initialiser le suivi visuel : ce traitement, réalisé par le module LOCPOST, n'est nécessaire que si la position du robot est très incertaine. Sinon, le module de suivi peut être activé à partir de la position prédite de l'amer dans l'image, position obtenue par reprojection ;
- suivi de l'amer, ici réalisé par la méthode différence de motifs ;
- enfin, asservissement de la caméra active pour centrer l'amer dans l'image et sélectionner la zoom pour avoir la résolution optimale.

5.4.2 Intérieur - navigation par suivi de primitives

Le deuxième contexte évoqué lors de la navigation en intérieur est la navigation du robot fondée sur une carte topologique. Pour cela, il faut une carte topologique, mais en plus, un module spécifique de génération de trajectoires qui exploite cette carte afin de générer automatiquement la trajectoire que le robot doit suivre. Cette trajectoire doit être exprimée sous la forme d'une séquence d'actions asservies ou contrôlées sur des capteurs (en anglais, *visual servoing* ou *sensor-based motion*), cela en fonction du but à atteindre par le robot, des informations présentes dans la carte topologique et des capteurs disponibles sur le robot (ici, la vision). Par exemple, une consigne du type *suivre le mur gauche*, sera exprimée comme *suivre l'arête entre le sol et le mur à gauche*.

Le générateur de trajectoires pourra générer un plan perceptuel adapté à ce mode de navigation, plan qui devra être exécuté par un ensemble de modules fonctionnels, en particulier par les modules de suivi visuel et de commande de la caméra active.

Dans l'architecture décisionnelle des démonstrateurs robotiques du LAAS-CNRS, entre la couche *planification* et la couche *modules fonctionnels*, la couche *supervision* va interpréter un plan perceptuel et va contrôler son exécution. Initialement, elle enverra une requête à un module de détection de primitives ou caractéristiques, situé au même niveau hiérarchique que le module de suivi (figure 1.7). La requête concernera la recherche dans l'image, d'une entité définie dans le plan perceptuel : soit une primitive spécifique (ligne mur/sol par exemple) soit un objet spécifique (affiche, porte ou panneau de signalisation). Le module de détection de primitives reportera par une réplique ou un bilan, le résultat de sa recherche à la couche *supervision* ou *contrôle d'exécution*. Si l'objet ou la primitive a été trouvé, des paramètres sur sa localisation dans l'image, seront exportés via un poster.

La couche *supervision* initialisera ensuite un module d'asservissement visuel et le module de suivi, afin de réaliser un mouvement asservi. Le module de suivi sélectionnera la méthode indiquée dans le plan pour le suivi de l'objet ou de la primitive, qui devient de ce fait, une cible ; il exportera l'état courant du suivi dans un poster qui sera lu par le module d'asservissement en charge des mouvements du robot.

Nous n'avons pas pu intégrer cette expérimentation complètement ; nous n'avons pas à ce jour dans le groupe, de module d'asservissement visuel assez robuste. Néanmoins, nous avons réalisé en temps réel, une séquence significative de suivi de primitives et/ou d'objets ; cette séquence de suivi est exécutée tandis qu'un opérateur pilote le robot de manière interactive.

Il ne s'agit pas réellement de l'exécution d'un plan perceptuel ; les enchaînements entre les phases de suivi, ont été ici programmés de manière ad hoc.

Cet exemple, pour illustrer un mode de navigation basée sur le suivi de primitives, est montré dans la figure 5.6:

- (a) cette expérimentation qui concerne la navigation visuelle d'un robot dans un couloir.
- (b) La première primitive à suivre est l'arête entre le sol et le mur à gauche du robot.

- (c) Ensuite, pendant que continue le suivi de cette ligne droite mur / sol, un détecteur d'évènements visuels (figure 4.21) comme celui montré dans la section 4.3.2, est utilisé pour détecter une arête verticale qu'il faudra suivre quand le suivi de l'arête mur /sol sera arrêté.
- (d) Soit parce que la première cible est perdue, ou plus simplement parce que cette la commutation a été planifiée, une seule cible est suivie ensuite : l'arête verticale.
- (e) Dans le même temps la caméra est commandée pour centrer la primitive suivie dans l'image.
- (f) Une nouvelle consigne dans le plan est dans ce cas de suivre une affiche. Comme pour tous les enchaînements entre cibles, cette affiche devrait être trouvée par un module de reconnaissance de poster, qui donnerait la fenêtre englobante du poster comme initialisation au module de suivi. Ici l'initialisation a été faite de manière interactive.
- (g) , (h) et (i). Le module de suivi ajuste les paramètres de la caméra (angles de champ de vue, site et azimut) pour maintenir la cible dans le champ de vue de la caméra, pendant que le robot suit une trajectoire droite. L'affiche pourrait être utilisée pour que le robot puisse s'assurer qu'il est dans le bon couloir
- (j) dans ce cas, le plan perceptuel contient un changement des paramètres de la caméra, qui tourne vers la droite pour aller détecter et suivre des arêtes sur le mur de droite.
- (k), (l) et (m) Ensuite la séquence continue avec le suivi des arêtes horizontales et verticales, arêtes mur / sol ou mur / porte.
- (n) un objet différent, non plan, est détecté, l'extincteur.
- (o) et (p) de par la nature de cet objet, il sera suivi par la méthode des contours actifs. Le module de suivi continue néanmoins de suivre les arêtes verticales ou horizontales qui ont été indiquées dans le plan.
- (q) le suivi de l'extincteur se poursuit jusqu'à un moment où l'objet touche le bas de l'image, produisant une perte du gradient de l'objet. C'est pourquoi le *snake* est déformé arbitrairement
- (r) ,(s) et (t) A partir de ce moment, le suivi continue sur à d'autres cibles, dans ce cas, un fois encore, des lignes droites, formées par les arêtes entre le sol et le mur, et entre les portes et le mur.

Même si dans cette séquence, les enchaînements ont été programmés explicitement, si certaines initialisations (en particulier, sur le poster et sur l'extincteur) sont assurées par l'opérateur, cette expérimentation est significative de l'intégration de plusieurs méthodes de suivi dans un même module, qui sélectionne en fonction de la nature de la cible, la méthode la plus adaptée. Dans cette séquence, nous exploitons les trois méthodes que nous avons développées dans nos travaux : suivi par corrélation 1D pour les lignes droites, suivi par différence de motifs pour le poster (l'apprentissage avait été fait au préalable), suivi par un contour actif pour l'extincteur, objet non plan, dont la silhouette dans l'image, se déforme du fait des déplacements du robot et de la modification en continue des paramètres de la caméra.

Il restera à intégrer notre module de suivi, avec les autres modules nécessaires dans ce mode de navigation ...

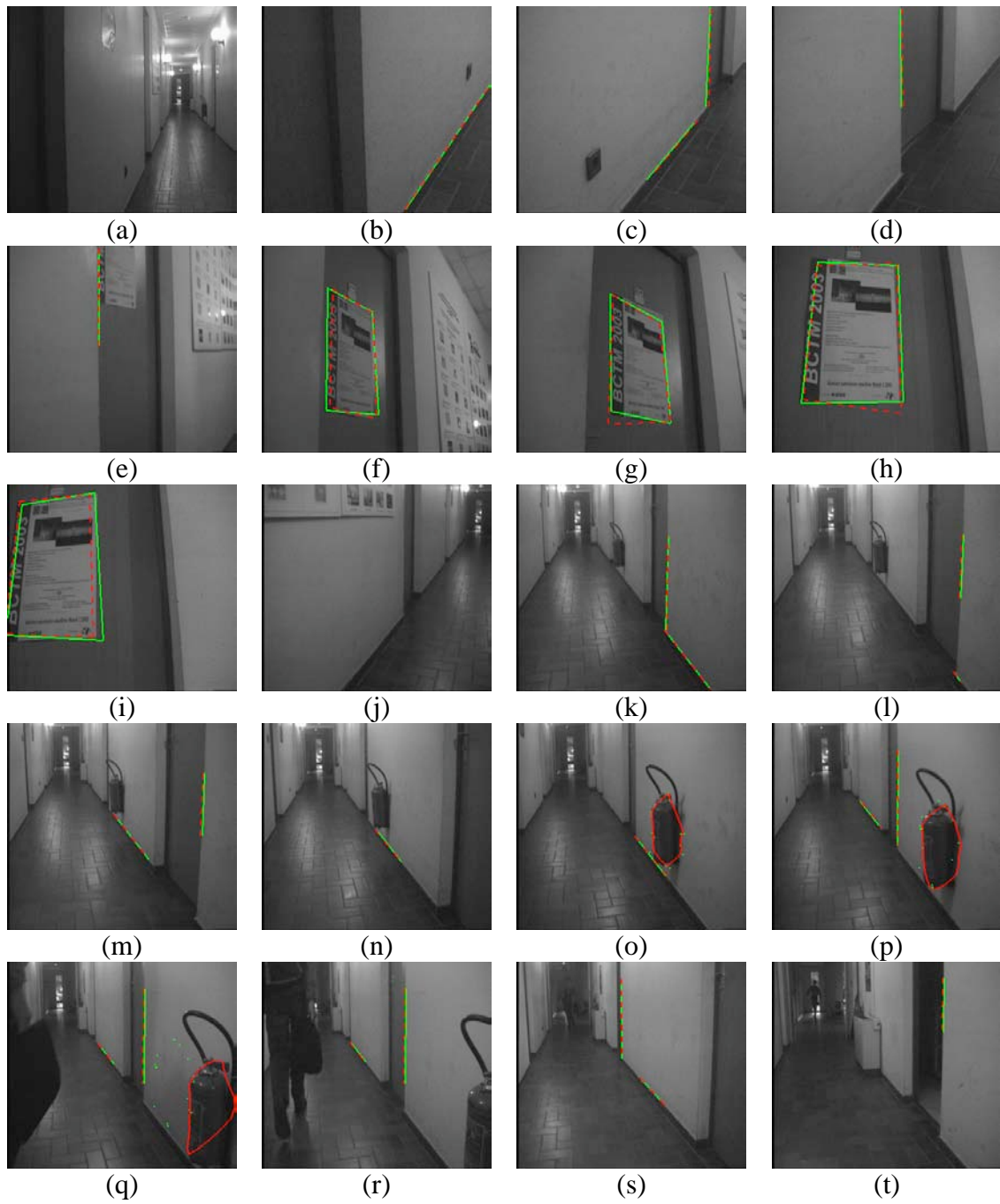


Figure 5.6. Navigation à partir du suivi des primitives.

5.4.3 Navigation en extérieur pour le suivi de routes ou chemins.

Comme nous l'avons mentionné au chapitre 1, la richesse de l'information contenue dans les images couleur acquises dans des environnements extérieur, est un facteur très important afin de réaliser l'extraction de régions dans cette sorte d'environnements. Afin de réaliser la navigation d'une machine en extérieur, en coopération avec G.Avina Cervantes, chargé d'intégrer cette expérimentation sur le démonstrateur DALA, nous nous sommes intéressés principalement à l'extraction et au suivi du chemin.

Dans ce contexte le module de coopération décrit dans la figure 1.7, sera un module chargé de l'extraction du chemin depuis une image couleur. Ce module sera décrit par la suite très rapidement.

Le problème de l'extraction des régions homogènes par rapport à une ou plusieurs caractéristiques distinctives, comme le niveaux de gris, la texture, la forme ou couleur, a été un sujet de recherche actif pendant les trois dernières décennies. Pour des images monochromatiques de nombreux algorithmes ont été développés. Cependant, le problème de la segmentation des images couleur, qui offrent beaucoup plus d'informations sur les objets présents dans un environnement extérieur en particulier, a fait l'objet d'une attention moindre de la communauté scientifique.

C'est à partir d'un algorithme de segmentation couleur, puis de la caractérisation des régions par des attributs de couleur et de texture, que l'extraction d'une route ou d'un chemin est faite. La méthode de segmentation développée, dans le groupe, notamment par G.Aviña-Cervantes [5], est basée sur un algorithme hybride, qui combine les techniques de croissance de régions et de *clustering*. Cet algorithme est indépendant du point de départ de la segmentation et de l'ordre d'adjacence des régions à fusionner et utilise un critère global de fusion. Les étapes les plus représentatives de l'algorithme sont :

- l'estimation des *clusters*, donc des différents seuils qui les séparent, à partir des histogrammes des trois composantes couleur choisies.
- l'étiquetage et la pré-classification des pixels en cellules connexes.
- la croissance des cellules en utilisant une connexité à 8 voisins.
- enfin, la fusion (élimination) des petites régions.

Les résultats de la segmentation sont montrés dans la figure 5.7. Dans l'implémentation actuelle de cet algorithme, l'espace I_1, I_2, I_3 (attributs non corrélés [88]), est utilisé pour la représentation de la couleur. Cependant, comme mentionné dans le chapitre 2, dans cet espace, le résultat pour la caractérisation des zones pour le suivi par contours actifs, n'est pas la meilleure option. Donc, les régions obtenues et plus précisément la région correspondante à la route, sera caractérisée dans l'espace de couleurs normalisées, et les valeurs moyennes et les variances des composantes couleur de la région route ou chemin, seront utilisées pour le suivi de cette zone par la méthode de contours actifs, comme mentionné dans la section 2.2.3.

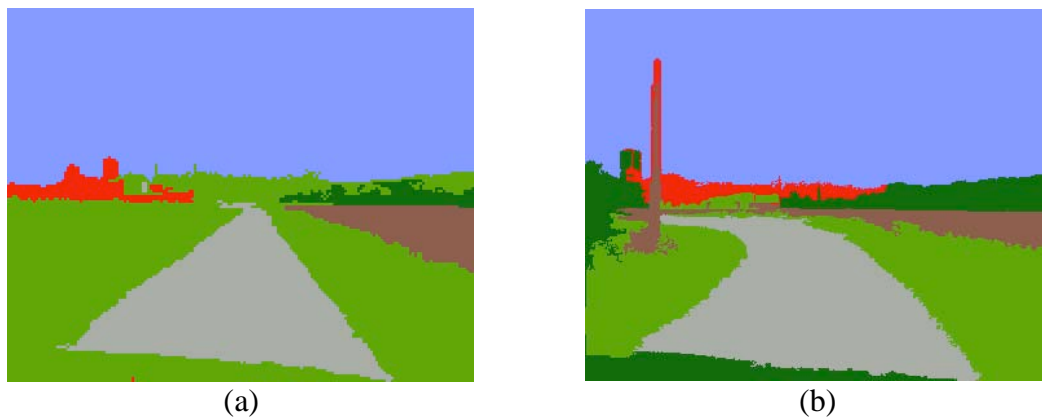


Figure 5.7. Résultats de la segmentation des régions dans une image couleur.

Dans ce cas, la coopération entre les modules d'extraction de la route dans les images couleur et le module de suivi se fait par moyen de posters : la route ou le chemin ayant une frontière déformable, elle sera suivie par un *snake*. Le module d'extraction envoie au second, d'une part les positions des points de contrôle pour l'initialisation du snake, d'autre part les paramètres qui caractérisent la région dans l'espace de couleurs normalisées (moyenne et variance).

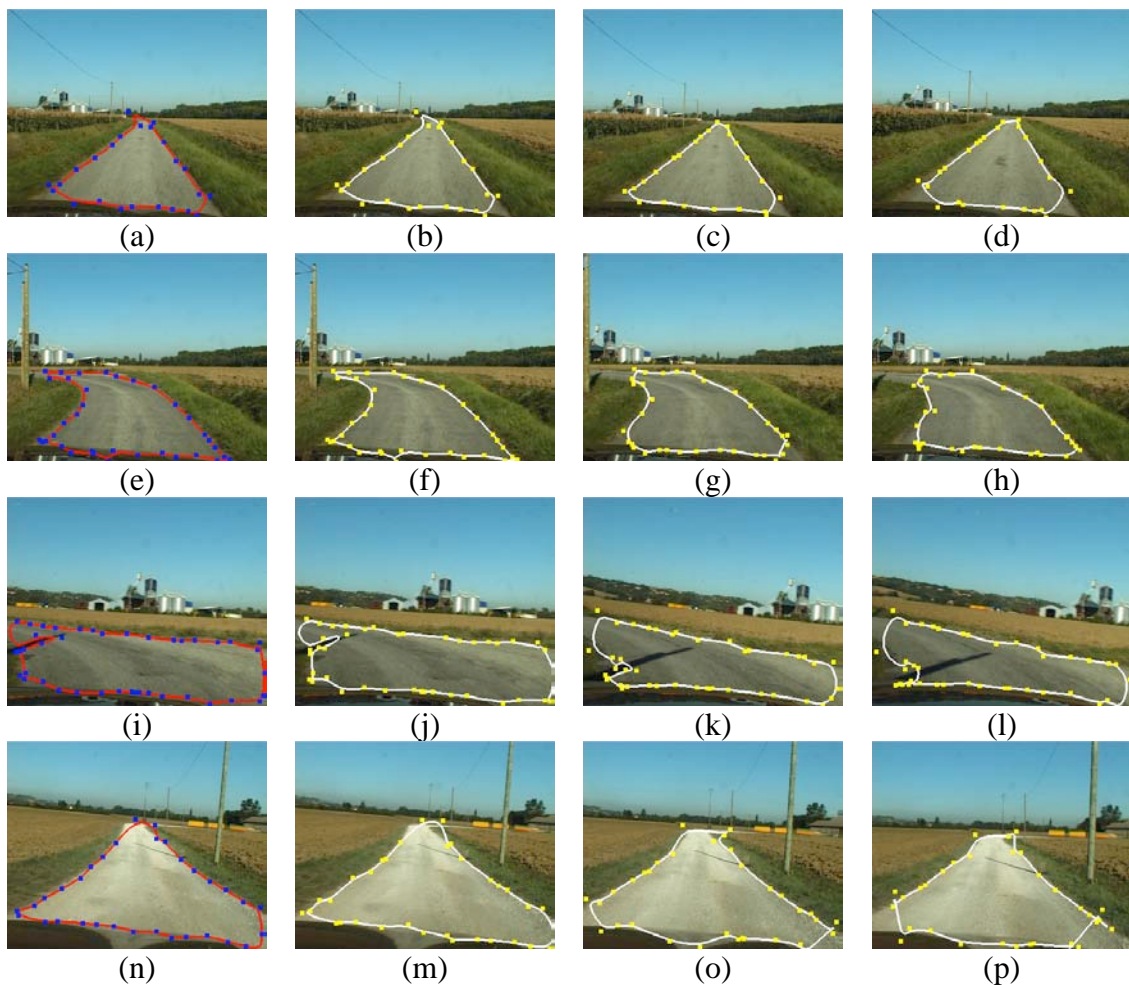


Figure 5.8. Navigation en extérieur

Le module de segmentation qui extrait la route, s'exécute environ à une fréquence de 1Hz, tandis que le suivi par *snake* sur des régions couleur, fonctionne environ à 10Hz. Toutes les secondes le module de suivi peut lire dans un poster, le contour de la route tel qu'il a été segmenté. Pour l'heure, ce contour sert à réinitialiser systématiquement le suivi, ce qui fait prendre un risque puisque le suivi fonctionne avec environ 10 images d'avance vis-à-vis de la segmentation. Il conviendrait d'avoir une stratégie plus adaptative, le suivi n'étant ré initialisé que s'il détecte une déformation suspecte du *snake* ou s'il existe trop d'écart entre l'état courant du snake et le résultat de la segmentation.

La figure 5.8 montre le suivi de route par contours actifs. Dans les images (a), (e), (i) et (n), sont montrées les initialisations du contour actif, données par les résultats de la segmentation de régions faites par le module d'extraction de route ; les autres images montrent le suivi pour différentes images dans la séquence. Comme on peut constater, la variation colorimétrique de la route est prise en compte au moment de l'initialisation du contour, afin d'éviter de le perdre du fait de changements d'illumination.

Rappelons que l'application de ces travaux concerne un projet franco-mexicain, financé par le Laboratoire Franco-Mexicain d'Informatique (LAFMI), qui traite de l'automatisation de machines agricoles.

5.4.4 Suivi d'obstacles en milieu routier.

Une dernière application du module de suivi a été réalisée, pour le suivi d'obstacles (voitures ou camions) en milieu routier, sur des séquences d'images acquises au LIVIC (Laboratoire des Interactions Véhicule-Infrastructure-Conducteur du LCPC et de l'INRETS) dans le cadre du projet ARCOS (Action de Recherche sur la Conduite Sécurisée)

Dans ce contexte le module d'extraction de caractéristiques est un module basé sur la détection d'obstacles depuis des images stéréo, développé dans le groupe par V. Lemonde. Ce module exportera la région détectée comme obstacle et ce sera à partir de la méthode de suivi par différences de motifs que le module de suivi fera sa tâche. La méthode de suivi par différences de motifs a été sélectionnée bien que les objets ne soient pas plans. Mais, du point de vue d'un conducteur, les voitures à l'avant du véhicule sont presque des cibles planes, et en plus, comme c'est une méthode rapide, elle permettra d'alerter rapidement du rapprochement dangereux d'un obstacle.

Dans la **Figure 5.9**, nous montrons, le suivi de l'arrière d'une voiture par la méthode de différences de motifs. Dans l'image (a) une voiture est détectée par un module de segmentation stéréo, et dans les images suivantes la partie arrière de ce véhicule est suivi. Comme on peut le constater, les différences de taille entre les images (a), (b) et (c), sont très importantes, et malgré cela, le suivi se déroule très bien. Dans cet exemple, l'approche du véhicule à l'avant, pourra alerter, soit le conducteur, soit le système automatique de navigation afin de prendre des distances par rapport à cet obstacle.

Le seul problème pour cette méthode est le temps nécessaire pour l'apprentissage de la matrice d'interaction.

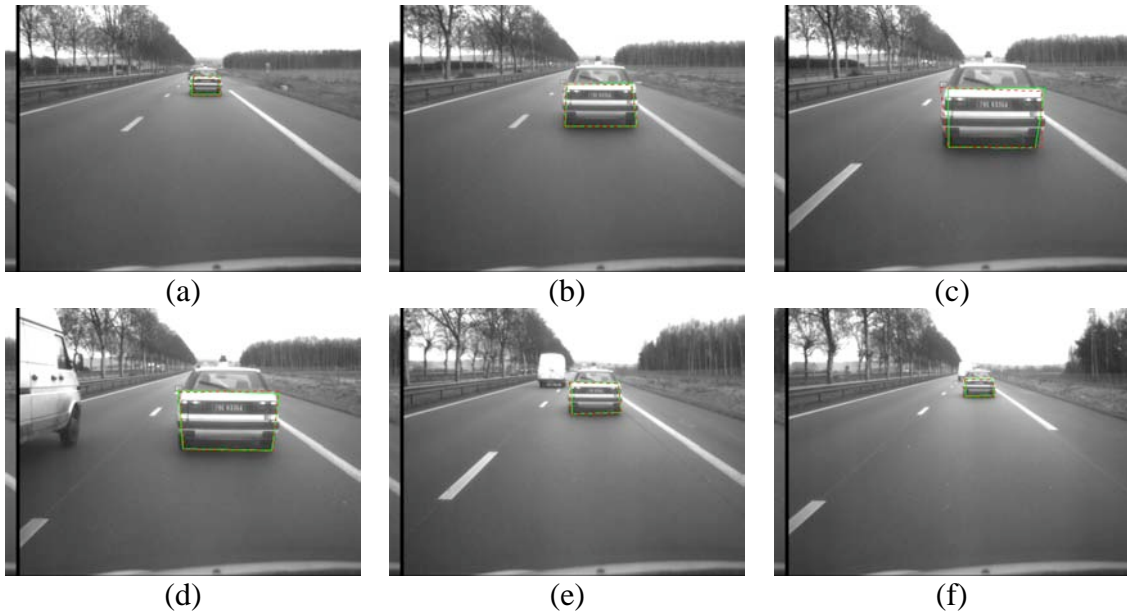


Figure 5.9. Suivi de l'arrière d'un véhicule par la méthode de suivi de motifs.

5.5 Conclusions

Dans ce chapitre, nous avons montré comment notre module de suivi d'objets a été intégré à l'intérieur de l'architecture LAAS du système embarqué sur les démonstrateurs dont disposent le groupe Robotique et Intelligence Artificielle.

Nous avons implémenté d'une manière simple, la commande des paramètres de la caméra par un contrôleur flou, dans ce contrôleur les résultats de l'ajustement de l'angle de champ de vue est utilisé pour limiter les vitesses définies dans la quantification des descripteurs linguistiques pour les variables de sortie, autant en site comme en azimut.

Un simulateur a été fait et montré, afin de tester la réponse de la caméra au contrôleur et leur accouplement avec les méthodes de suivi, nous montrons dans ce simulateur un expérimentation sur un environnement d'intérieur et le suivi des affiches, provenant d'un plan perceptuelle. Ce style d'environnement sera utilisé en réalité pour la localisation métrique d'un robot, cette expérimentation est en cours.

Deux autres expérimentations on été faits, une pour la navigation d'un robot mobile avec les suivi des primitives, ou on montre différentes méthodes que peuvent être utilisés, une expérimentation en environnements d'extérieurs pour la navigation d'une machine agricole, laquelle doit trouver est suivre leur chemin, et finalement une expérimentation pour le suivi d'obstacles en milieu routier, notamment l'arrière d'un véhicule.

Conclusion générale et perspective

Nous avons présenté les travaux développés sur les applications de la Vision Dynamique à la Navigation d'un robot mobile.

Nos contributions, concernent l'utilisation de fonctionnalités visuelles dans le système embarqué à bord d'un robot mobile. Les travaux en ce domaine se focalisent généralement sur un type d'environnement ; pour notre part, nous avons analysé des applications de nos travaux sur la Vision Dynamique, dans deux types d'environnement, intérieur et extérieur semi-structuré. En liaison avec JB.Hayet, qui a soutenu sa thèse sur la navigation fondée sur des amers, nous nous sommes d'abord intéressé à la navigation en milieu intérieur, avec deux approches différentes qui sont : a) la navigation métrique basée sur une localisation du robot vis-à-vis d'un repère global de l'environnement, et b) la navigation fondée sur une carte topologique. Dans les deux cas, nous avons traité le suivi d'amers visuels pendant les déplacements du robot. Ensuite, en coopération avec deux autres doctorants, nous avons aussi considéré la navigation en environnements extérieurs semi structurés, plus précisément la navigation de machines sur un réseau de routes ou chemins. Avec G.Avina, chargé de l'analyse de scènes à partir d'images couleur, nous avons traité du suivi visuel des bords d'une route ou d'un chemin sur lequel un robot doit se déplacer. Avec V.Lemonde, qui traite de la détection d'obstacles pour un véhicule circulant sur une route, nous avons considéré le suivi visuel d'un autre véhicule.

La caractéristique commune à ces travaux est la nécessité d'intégrer dans la couche fonctionnelle du système embarqué sur le robot, plusieurs fonctionnalités visuelles, ayant des temps de réponse différents ;

- les unes réalisent une analyse globale de l'image, et génèrent des connaissances : détection d'amers à des fins de localisation, détection de la surface *route* ou *chemin* sur laquelle le robot doit se avancer ou détection d'obstacles à éviter. Le temps de réponse typiquement est de 1s.
- les autres réalisent le suivi visuel d'entités de type amers, obstacles ou bords de route. Pour optimiser ces traitements, l'image est analysée localement dans des régions d'intérêt et la caméra peut être contrôlé pour orienter l'axe optique vers la cible suivie et choisir la meilleure résolution. La fréquence de traitement est typiquement entre 10 et 20Hz.

Nous avons présenté dans cette thèse, notre contribution sur ces fonctionnalités de suivi visuel. En sus d'être **rapides** pour assurer une bonne réactivité du système, ces fonctions doivent aussi être **robustes**, afin de minimiser les ruptures de suivi, et **génériques**, afin de pouvoir suivre plusieurs types de cible, selon le contexte. Ces deux dernières contraintes ne peuvent être satisfaites que par l'intégration de plusieurs méthodes de suivi, chacune ayant des caractéristiques propres en terme de représentations de la cible et de l'observation et en terme des méthodes mises en œuvre pour générer des hypothèses sur la position de la cible dans l'image, puis pour mesurer la pertinence de ces hypothèses.

Dans le chapitre 2 , nous avons présenté les quatre méthodes intégrées dans notre module de suivi visuel : les deux premières sont adaptées pour le suivi de contours pour des objets structurés (corrélation 1D) et pour des objets de forme quelconque (*snakes*) ; les deux autres permettent de suivre des régions texturées, représentées soit par un ensemble de points, soit par un motif. Nous avons implémenté trois de ces méthodes, et contribué plus particulièrement sur les points suivants :

- suivi par des contours actifs : rajout de forces internes pour garantir une bonne répartition des points de contrôle le long du contour ; rajout de forces pour imposer des contraintes de forme sur les points de contrôle du type 'sommet' ou 'segment de droite' ; et enfin, en coopération avec G.Avina, méthode dédiée pour traiter du suivi des bords d'une région de couleur uniforme.
- suivi par différence de motifs : introduction d'un repère lié à la cible, afin d'avoir une matrice d'interaction indépendante de l'initialisation, et par la même, la possibilité de combiner plusieurs résultats d'apprentissage.

Dans le chapitre 3 , ces méthodes ont été comparées et évaluées de manière qualitative. Nous avons estimé leurs domaines de validité concernant (1) le type de cibles qu'elles permettent de suivre, (2) le type d'environnement dans lequel cette cible est suivie, et (3) la robustesse vis-à-vis des mouvements apparents de la cible dans la séquence d'images.

Pour chaque méthode, nous avons également analysé deux traitements nécessaires : (a) la fonction d'initialisation du suivi (détection initiale de la cible dans la première image) et (b) une fonction d'évaluation, chargée de mesurer la qualité du suivi, et surtout, de détecter les ruptures ou pertes de cible.

Un grand nombre de validations expérimentales ont été nécessaires pour qualifier ces méthodes et pour vérifier la pertinence des mesures sur leur comportement.

Dans le chapitre 4 , nous abordons des aspects de contrôle perceptuel. Si une machine est dotée d'un module de suivi visuel et d'une caméra active, comment les exploiter au mieux ?

Nous avons tout d'abord poursuivi des travaux sur la planification, menés initialement par V.Ayala Ramirez ; il s'agit de planifier les configurations d'une caméra active le long d'une trajectoire planifiée par ailleurs, afin d'optimiser la perception sur des amers appris au préalable. Vis-à-vis de la méthode initiale, nous avons considéré des contraintes de visibilité sur les amers, prenant en compte leurs orientations et dimensions, puis pour traiter la planification, nous avons proposé des alternatives pour le calcul de l'utilité des cibles en fonction des paramètres de la caméra. La recherche exhaustive proposée dans les travaux précédents a été remplacée par une solution analytique. Enfin, nous avons proposé la génération du plan perceptuel en fonction des

combinaisons de tous les amers visibles, afin de résoudre le plan dans sa totalité par la programmation dynamique, en minimisant les saccades.

Nous avons ensuite considéré le contrôle de l'exécution d'un tel plan perceptuel. Deux types d'événements doivent être traités par le système de contrôle du module de suivi :

- d'une part, l'enchaînement entre deux phases de suivi (passage d'un amer *A* à un amer *B* par exemple), avec éventuellement une commutation dans la méthode de suivi,
- d'autre part, l'échec d'une procédure de suivi, du fait d'une occultation ou d'une perte de la cible.

Nous avons proposé et mis en œuvre différentes stratégies simples pour traiter ces événements, sans nuire à la réactivité de système d'ensemble. Dans le cas général, si la procédure de reprise d'échec est prise en défaut, il conviendra de faire appel à une autre fonctionnalité qui analyse l'image de manière globale.

Le chapitre 5 est justement consacré à la coopération entre différentes fonctionnalités visuelles : avec les doctorants cités ci-dessus, nous avons essayé d'intégrer à bord des démonstrateurs du groupe Robotique et Intelligence Artificielle, notre module de suivi visuel avec des modules de détection et localisation d'amers visuels, de route ou d'obstacles. Certaines de ces intégrations sont en cours et devront être finalisées. Nos objectifs dans tous les cas consistent à valider des interactions simples entre les modules :

- initialisation du suivi à partir de l'entité localisée dans l'image par le module de détection.
- Réinitialisation du suivi par le module de détection, réalisée de manière systématique à chaque localisation de la cible par le module de détection, ou seulement en cas de perte de la cible par le module de suivi.
- Réciproquement, prise-en-compte dans le module de détection de la position dans l'image, de la cible suivie, afin de simplifier l'analyse globale de l'image.

Nous avons également traité dans ce chapitre, de la commande d'une caméra active en fonction de la position de la cible dans l'image ; pour s'adapter aux faibles capacités offertes sur la caméra pour la commande des rotations en site et azimut de du zoom, nous avons adapté une approche de la commande floue. Des résultats expérimentaux nombreux ont permis de valider l'approche mise en œuvre.

En attendant de valider certaines des interactions entre toutes les fonctionnalités embarquées, nous avons présenté plusieurs résultats de suivi, obtenus

- dans un premier temps, en simulation, afin de valider toutes les approches proposées pour le suivi de cibles, mais aussi pour la commande de la caméra.
- en ligne et en temps réel, en milieu intérieur, depuis la version embarquée de notre module sur le robot DILIGENT.
- hors ligne sur des séquences enregistrées au LAAS en milieu extérieur, en attendant l'intégration de notre module de suivi sur le robot DALA (en cours).
- hors ligne sur des séquences enregistrées au LIVIC en milieu routier.

Enfin, nous souhaitons terminer en évoquant des travaux que nous pourrions mener dans le futur sur la thématique présentée dans cette thèse. Tout d'abord, de nombreuses améliorations devraient être apportées aux modules logiciels que j'ai développés : augmentation de la base de cibles prises en compte afin d'ajuster les paramètres de la sélection de la méthode de suivi visuel, incorporation de nouveaux contextes, amélioration de la méthode de suivi par corrélation 1D..., incorporation d'autres mesures de complexité, concernant par exemple la structure de la scène, ou l'intérieur de la cible....

A plus long terme, plusieurs études pourraient être poursuivies, notamment sur les points suivants :

L'intégration de fonctions de prédiction semblent le plus nécessaire pour prendre en compte des cibles mobiles avec des dynamiques importantes. Nous avons négligé cet aspect, traité par d'autres doctorants. Par exemple, T.Sentenac a proposé des méthodes d'estimation fondées sur le filtrage de Kalman, mais capables de gérer plusieurs modèles possibles sur la trajectoire ou la dynamique des cibles. Intégrer dans un même système ces méthodes et le suivi visuel est un objectif prioritaire ; une application intéressante étant la surveillance d'environnements dynamiques.

En particulier, l'estimation de la dynamique des objets mobiles détectés dans les séquences d'images acquises depuis un véhicule ou un robot en mouvement, est toujours un sujet difficile, qui pourrait trouver des applications importantes pour prédire les collisions possibles entre véhicules.

Dans le cas d'échec, la génération des plusieurs hypothèses afin de reprendre l'activité, pourra être intégrée et testée à partir d'un filtrage particulière.

Le suivi d'objets déformables comme les silhouettes des personnes, ou la main d'un opérateur, fait l'objet de la thèse de L.Brethès. Nous pourrions aussi évaluer nos méthodes sur ces applications.

Pour le cas du suivi de plusieurs objets dans des contextes plus difficiles (contrôle de trafic, vidéo surveillance...), l'exécution en parallèle de deux ou plus méthodes de suivi fondées sur différentes caractéristiques, pourrait également être étudiée : comment fusionner leurs résultats ?

Finalement, nous pourrions analyser comment sur un robot, intégrer les résultats de nos modules de vision avec ceux obtenus avec les données d'autres capteurs, comme les caméras omnidirectionnelles, la vision stéréo ou la télémétrie laser.

Références Bibliographiques

- [1] Adachi, Y., M. Asada, and T. Nakamura, "Correlation-based visual tracking enhanced by affine motion description", in *Proc. of IAPR Workshop on Machine Vision Applications '96*, pp. 75-78, 1996.
- [2] Alami R., R. Chatila, S. Fleury, M. Ghallab and F. Ingrand, "An Architecture for Autonomy", in *International Journal of Robotics Research*, vol. 17, No. 4, pp. 315-337, 1998.
- [3] Aloimonos Y., "Purposive active vision," in *CVGIP: Image Understanding*, vol. 56, No. 1, pp. 840-850, August 1992.
- [4] Amini, A. A., T.E. Weymouth, R.C. Jain, "Using Dynamic Programming for Solving Variational Problems in Vision", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, Sep. 1990.
- [5] Aviña, G., M. Devy, R., Murrieta, "Road Detection for Robot Navigation", in *Proceedings of International Symposium on Robotics and Automation*, Toluca, Mexico, Sep., 2002.
- [6] Ayala-Ramirez, V. C., *Fonctionnalités Visuelles sur des Scènes Dynamiques pour la Robotique Mobile*, Thèse de Doctorat, Université Paul Sabatier, Toulouse, 2000.
- [7] Ayala-Ramírez, V., C. Parra and M. Devy, "Active tracking based on Hausdorff matching", in *Proc. of the 15th Int. Conf. on Pattern Recognition (ICPR'2000)* vol. 4, pp. 706-709, Barcelona, Spain, Sep. 2000, IAPR.
- [8] Bajcsy R., "Active Perception", in *Proceedings of the IEEE, Special issue on Computer Vision*, Vol. 76, No. 8, August 1988.
- [9] Balkenius, C. and L. Kopp, "Visual Tracking and Target Selection for Mobile Robots", in *Proceedings of EUROBOOT'96*, 1996.
- [10] Bar-Shalom, Y., X. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*, John Wiley, 2001.
- [11] Barreto, J. P., P. Peixoto, J. Batista and H. Araujo, "Evaluation of the Robustness of Visual Behaviours", in *Robust vision for vision-based control motion*, M. Vincze and G.D. Hager editors, Wiley-IEEE Press, February, 2000.
- [12] Bascle B., P. Bouthemy, R. Deriche, and F. Meyer, "Tracking complex primitives in an image sequence", *Technical Report 2428*, INRIA, Sophia-Antipolis, France, December 1994.

- [13] Bascle, B. and R. Deriche, "Region tracking through image sequences", in Proc. *5th Intl. Conf. on Computer Vision*, Cambridge, MA, pp. 302-307, 1995.
- [14] Baumberg, A.M., *Learning Deformable Models for Tracking Human Motion* Ph.D. Thesis, School of Computer Science, University of Leeds, October, 1995.
- [15] Betke, M., and L. Gurvits, "Mobile Robot Localization Using Landmarks", in *IEEE Trans. On Robotics and Automation*, vol. 13, No. 2, April, 1997.
- [16] Bhanu, B., L. Yingqiang, G. Jones and J. Peng, "Adaptive target recognition", in *Machine Vision and Applications*, Vol. 11, No. 6, pp. 289-299, 2000.
- [17] Blaer, P, and P. Allen, "Topological mobile robot localization using fast vision techniques", in *Proceeding of International Conference on Intelligent Robots and Systems IROS'02*, pp. 1031-1036, Lausanne, Switzerland, 2002.
- [18] Blake, A. and A. Yuille (editors), "Active Vision", MIT press, November, 1992.
- [19] Blake, A. and M. Isard, "Active Contours", Springer-Verlag, London, 1998.
- [20] Borges, G.A., M.-J. Aldon, "A Split-and-Merge Segmentation Algorithm for Line Extraction in 2-D Range Image" in Proc. of *ICPR 2000*, pp. 1441-1444, 2000.
- [21] Bothemy, P., "A Maximun Likelihood Framework for Determining Moving Edges", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, No. 5, pp. 499-511, May. 1989.
- [22] Buenaposada, J. M. and L. Baumela, "Real-time tracking and estimation of plane pose", in *Proceedings of ICPR'02*, IEEE, vol. II, pp. 697-700, 2002.
- [23] Chatila, R. and S. Lacroix, "Adaptive navigation for autonomous mobile robots", in Proc of *International Symposium on Robotics Research (ISRR'95)*, Munich, October 21-24, 1995.
- [24] Cheng, H. D., X. H. Jiang and Y. Sun and Jingli, "Color image segmentation: advances and prospects", in *Pattern Recognition*, vol. 34, No. 12, pp. 2259-2281, 2001.
- [25] Chihman, V.N., V. M. Bondarko, M. V. Danilova, Y. E. Shelepin, A. Goluzina, "Comparison of measures of complexity of visual objects", in *25th European Conference on Visual Perception (ECVP'2002)*, August 25-29, Glasgow, Scotland, 2002.
- [26] Choset, H. and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization", in *IEEE Trans. on Robotics and Automation*, vol. 17, No. 2, pp. 125-137, April 2001.
- [27] Chroust S., J. Barreto, H. Araujo, and M. Vincze, "Comparison of control structures for visual servoing", in *Proceedings of ICAR'2001*, pp. 235-240, Budapest, August, 2001.
- [28] Clady, X., F. Collange, F. Jurie, P. Martinet, "Object Tracking with a Pan Tilt Zoom Camera application to car driving assistance", in Proc. of *ICRA 2001*, pp. 1653-1658, 2001.
- [29] Crowley J.L., "Mathematical fundation of navigation and perception for an autonomous mobile robot", in *Reasoning with Uncertainty in Robotics (L. Dorst ed.)*, Springer-Verlag, 1996.

- [30] Crowley, J.L., and F. Berard, "Multi-modal tracking of faces for video communication", in Proc. of *Computer Vision and Pattern Recognition* (CVPR'1997), San Juan, Puerto Rico, pp. 640-645, June 17-19, 1997.
- [31] Dao N.X., BJ.You, SR.Oh and M. Hwangbo, "Visual Self-Localization for Indoor Mobile Robots Using Natural Lines", in Proc. of the 2003 *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems IROS'2003*, pp. 1252-1257, Las Vegas, Nevada, October, 2003.
- [32] Davison, A. J. and D. W. Murray, "Mobile robot localization using active vision", in *European Conference on Computer Vision (ECCV'98)*, June 2-6, 1998.
- [33] de Berg, M., M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, second edition, Springer-Verlag, Berlin, 2000.
- [34] Dedeoglu, G., M.J. Mataric and Gaurav S. Sukatme, "Incremental, on-line topological map building with a mobile robot", in Proc. of *Mobile Robots XIV*, SPIE vol. 3838, pp. 129-139, Boston, MA, 1999.
- [35] Dellaert F., C. Thorpe, and S. Thrun, "Super-Resolved Texture Tracking of Planar Surface Patches", in Proc. of *IEEE/RSJ International Conference on Intelligent Robotic Systems*, October, 1998.
- [36] Demengel, G., and J. P., Pouget, *Mathématiques des Courbes et des Surfaces : Modèles de Bézier, des B-Splines et des NURBS*, Ellipses, Paris, 1998.
- [37] DeSouza, G. N. and Avinash C. Kak, "Vision for Mobile Robot Navigation: A Survey", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, No. 2, pp. 237-267, February, 2002.
- [38] Douze M., V. Charvillat and B. Thiesse, « Comparaison et intégration de trois algorithmes de suivi de motifs plans », in Proceedings of *ORASIS'2003*, pp. 221-230, Gerardmer, France, May, 2003.
- [39] Drummond, T. and R. Cipolla, "Real-time tracking of complex structures with on-line camera calibration", in *Image and Vision Computing*, vol. 20, No. 5-6, pp. 427-433, 2002.
- [40] Drummond, T. and R. Cipolla, "Real-Time Visual Tracking of Complex Structures", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, No. 7, pp. 932-946, July, 2002.
- [41] Fagiani, C., *An Evaluation of Tracking Methods for Human-Computer Interaction*, Senior Thesis, Computer Science Department, Boston College, 2002.
- [42] Fernández, J. and A. Casals, "Autonomous Navigation in Ill-Structured Outdoor Environments", in *Proceeding of International Conference on Intelligent Robots and Systems IROS'97*, pp. 395-400, Grenoble, France, September, 1997.
- [43] Findlay, J. M. and I. D. Gilchrist, *Active Vision - The Psychology of Looking and Seeing*, Oxford University Press, London, august, 2003.
- [44] Fischler, M. A., and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", in *Communications of the ACM*, vol. 24, No. 6, pp. 381-395, June, 1981.

- [45] Gaspar, J., E. Grossmann, and J. Santos-Victor, "Interactive Reconstruction from an Omnidirectional Image", *9th International Symposium on Intelligent Robotic Systems (SIRS2001)*, Toulouse, France, July 2001.
- [46] Gevers, T., S. Ghebreab, A. Smeulders, "Color Invariant Snakes", in *Proceedings of the Ninth British Machine Vision Conference*, pp. 578-588, Southampton, UK, 1998.
- [47] Gonzalez-Barbosa, J. J. and S. Lacroix, "Rover Localization in Natural Environments by Indexing Panoramic Images", in Proc. of *IEEE International Conference on Robotics and Automation (ICRA '02)*, Washington, D.C., 2002.
- [48] Hager, G. D. and P. N. Belhumeur, "Efficient Region Tracking With Parametric Models of Geometry and Illumination", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, No. 10, pp. 1025-1039, October 1998.
- [49] Harris, C. and M. Stephens, "A combined corner and edge detector", in *Proceeding of the 4th Alvey Vision Conference*, pp. 147-151, 1988.
- [50] Hashima M., F. Hasegawa, S. Kanda, T. Maruyama, and T. Uchiyama, "Localization and Obstacle Detection for a Robot for Carrying Food Trays", in *Proceeding of International Conference on Intelligent Robots and Systems IROS'97*, pp. 345-351, Grenoble, France, September, 1997.
- [51] Hayet, J.B., Contribution à la navigation d'un robot mobile sur des amers visuels textures dans un environnement structure, Thèse de doctorat, Université Paul Sabatier, Toulouse, 2003.
- [52] Huttenlocher, D. P. et al, "Visually guided navigation by comparing two dimensional edge images", Technical. Report, TR-94-1407, Stanford University, Stanford, California, January, 1994.
- [53] Isard. M and A. Blake, "CONDENSATION -- conditional density propagation for visual tracking", in *Int. J. Computer Vision*, vol. 29, No 1, pp. 5--28, 1998
- [54] Ishiguro, H. and S. Tsuji, "Image-based memory of the environment", in *Proceeding of International Conference on Intelligent Robots and Systems IROS'96*, vol. 2, pp. 634-639, Osaka, Japan, November 4-8, 1996.
- [55] Ivins, J. and J. Porril, "Active Region Models for Segmenting Textures and Colours", in *Image and Vision Computing*, vol. 13, No. 5, pp. 431-438, 1995.
- [56] Ivins, J. and J. Porrill, "Constrained Active Region Models for Fast Tracking in Color Image Sequences", in *Computer Vision and Image Understanding*, vol. 72, No. 1, pp. 54-71, Oct, 1998.
- [57] Jennings G.A., *Modern Geometry with Applications*, Springer-Verlag, New York, 1997.
- [58] Jepson, A. D., D. J. Fleet and T. F. El Maraghi, "Robust online appearance models for visual tracking", in Proceeding of *IEEE Conf. on Computer Vision and Pattern recognition CVPR'01*, vol. 1, pp. 415-422, Kauai, 2001.
- [59] Jung, I.K., S. Lacroix, "A robust interest points matching algorithm", in Proc. of *8th International Conference on Computer Vision (ICCV'2001)*, Vancouver (Canada), vol. II, pp.538-543, July 9-12, 2001.

- [60] Jurie, F. and M. Dhome, "A simple and efficient matching algorithm", in *Proceeding of ICCV'99*, pp. 265-275, 1999.
- [61] Jurie, F. and M. Dhome, "Real time tracking of 3D objects: an efficient and robust approach", in *Pattern Recognition*, vol. 35, No. 2, pp. 317-328, 2002.
- [62] Jurie, F. and M. Dhome, "Hyperplane Approximation for Template Matching", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, No. 7, pp. 996-1000, July 2002.
- [63] Kass, M., A. Witkin, and D. Terzopoulos, "Snakes: Active Contours Models", in *Int'l J. Computer Vision*, vol. 1, pp. 321-331, 1988.
- [64] Kim, H. and K. S. Hong, "Robust Image Mosaicing of Soccer Videos using Self-Calibration and Line Tracking", in *Pattern Analysis & Applications*, vol. 4, pp. 9-19, 2001.
- [65] Kiryati, N. and A. M. Bruckstein, "What's in a Set of Points", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, No. 4, pp. 496-500, April, 1992.
- [66] Klinger A. and N. A. Salingaros, "A pattern measure", in *Environment and Planning B: Planning and Design*, vol. 27, pp. 537-547, 2000.
- [67] Koller, D., J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning" in *Proc. 3rd European Conf. on Computer Vision (ECCV'94)*, Stockholm, vol. 1, pp. 189-196, May, 1994.
- [68] Kristensen, S. and P. Jensfelt, "An Experimental Comparison of Localisation Methods, the MHL Sessions", in *Proceeding of International Conference on Intelligent Robots and Systems IROS'03*, pp. 992-997, Las Vegas, Nevada, October, 2003.
- [69] Kruppa, H., M. Spengler and B. Schiele, "Context driven model switching for visual tracking", in *Proceedings of 8th Int. Symposium on Intelligent Robotic Systems SIRS'00*, 2000.
- [70] Labayrade R., C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner", in *IEEE ICAR'2003*, Coimbra, Portugal, June 30 - July 3, 2003.
- [71] Lacroix, S., *Stratégies de perception et de déplacement pour la navigation d'un robot mobile autonome en environnement naturel*, Doctorat, Université Paul Sabatier, Toulouse, 13 April 1995, 2000.
- [72] Livatino S. and C.B. Madsen, "Optimization of robot self-localization accuracy by automatic visual-landmark selection", in *Scandinavian Conference on Image Analysis (SCIA)*, 1999.
- [73] McLauchlan, P. F. and J. Malik, "Vision for Longitudinal Vehicle Control", in *Proceedings of the Eighth British Machine Vision Conference (BMVC'97)*, 1997.
- [74] Madsen C.B. and C.S. Andersen, "Optimal landmark selection for triangulation of robot position", in *Robotics and Autonomous Systems*, vol. 23, No. 4, pp. 277-292, 1998.
- [75] Marin-Hernandez, A., and H.V. Rios-Figueroa, "Eels: Electric Snakes", in *Computación y Sistemas*, vol. 2, no. 2-3 pp. 87-94, 1999.

- [76] Marr, D. "Vision", W H Freeman & Co., June 1982.
- [77] Mata, M., J. M. Armingol, A. de la Escalera and M. A. Salichs, "Using learned visual landmarks for intelligent topological navigation of mobile robots", in Proceedings of *IEEE international Conference on Robotics and Automation (ICRA'03)*, Taipei, Taiwan, September 14-19, 2003.
- [78] Menezes, P. L. Brethes, F. Lerasle, P. Danes, J.Diaz, "Visual tracking of silhouettes for human-robot interaction", in Proc. of *11th International Conference on Advanced Robotics (ICAR'2003)*, Coimbra (Portugal), vol. 2, pp. 971-976, June 30 – July 3, 2003.
- [79] Montgomery, D. C., E. A. Peck, and G. Geoffrey, *Introduction to Linear Regression Analysis*, 3rd Edition, Wiley, April, 2001.
- [80] Moutarlier, P., *Modélisation autonome de l'environnement par un robot mobile* Thèse de Doctorat, Université Paul Sabatier, Toulouse, Octobre 1991.
- [81] Murrieta-Cid, R. "Contribution au d'veloppement d'un système de vision pour robot mobile d'extérieur", Ph.D. Thesis, Institut National Polytechnique de Toulouse, Toulouse, France, Nov. 1998.
- [82] Nashashibi, F., Perception et modélisation d'un environnement tridimensionnel pour la navigation autonome d'un robot mobile, Thèse de Doctorat, Université Paul Sabatier, Toulouse, Janvier, 1993.
- [83] Neira J., M.I. Ribeiro, and J.D. Tardos, "Mobile robot localization and map building using monocular vision", in *Symposium on Intelligent Robotics Systems (SIRS'97)*, 1997.
- [84] Neto, G. C. Costelha, and P. Lima, "Topological Navigation in Configuration Space Applied to Soccer Robots", in *Robocup 2003 International Symposium*, Padova, Italy, 2003.
- [85] Nickels, K., S. Hutchinson, "Estimating uncertainty in SSD-based feature tracking", in *Image and Vision Computing*, vol. 20, pp. 47-58, 2002.
- [86] Niimi, M., H. Noda and E. Kawaguchi, "An image embedding in image by a complexity based region segmentation method", in Proc. of *International Conference on Image Processing (ICIP'97)*, vol. 3, pp. 74-77, October, 1997.
- [87] Ng, J. and S. Gong, "Learning Pixel-Wise Signal Energy for Understanding Semantics", in Proc. of *BMVC'01*, pp. 695-704, 2001.
- [88] Ohta, Y., T. Kanade, and T. Sakai, "Color information for region segmentation", in *Computer Vision, Graphics and Image Processing*, vol. 13, pp. 222-241, 1980.
- [89] Peters, R. A. II, and R. N. Strickland, "Image Complexity Metrics for Automatic Target Recognizers", in *Proceedings of Automatic Target Recognizer System and Technology Conference*, Silver Spring, USA, October, 1990.
- [90] Rasmussen C. and G. Hager, "Probabilistic data association methods for tracking complex visual objects, in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, No.6, pp. 560-576, June, 2001.
- [91] Restelli M., D. G. Sorrenti, and F. M. Marchese, "A robot localization method based on Evidence Accumulation and Multi-Resolution", in *Proceeding of*

International Conference on Intelligent Robots and Systems IROS'02, pp 415-420, Lausanne, Switzerland, 2002.

- [92] Ross, T.J., *Fuzzy Logic with Engineering Applications*, McGraw-Hill, 1995.
- [93] Sapiro, G., "Color Snakes", *Computer Vision and Image Understanding*, vol. 68, No. 2, November, pp. 247-253, 1997.
- [94] Shearer, K., K. D. Wong and S. Venkatesh, "Combining multiple tracking algorithms for improved general performance", in *Pattern Recognition*, Vol. 34, No.6. pp. 1257-1269, 2001.
- [95] Shen, D. and C. Davatzikos, "An Adaptive-Focus Deformable Model Using Statistical and Geometric Information", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 906 -913, August, 2000.
- [96] Sherrah J. and S. Gong, "Continuous Global Evidence-Based Bayesian Modality Fusion for Simultaneous Tracking of Multiple Objects", In Proc. *IEEE International Conference on Computer Vision ICCV'2001*, Vancouver, Canada, pp. 42-49, 2001.
- [97] Shi J., and C. Tomasi, "Good Features to Track", in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, USA, June, 1994.
- [98] Simmons, R. and S. Koenig, "Probabilistic Robot Navigation in Partially Observable Environments", in Proceedings of the *International Joint Conference on Artificial Intelligence*, pp. 1080-1087, 1995.
- [99] Smith, R. C. and P. Cheeseman "On the representation and estimation of spatial uncertainty" in *International Journal of Robotics Research*, vol. 5, No. 4, pp. 56-68, 1986.
- [100] Smith S.M. and J.M. Brady, "ASSET-2: Real-Time Motion Segmentation and Shape Tracking", in *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, No. 8, pp. 814-820, August 1995.
- [101] Spengler, M, and B. Schiele, "Towards robust multi-cue integration for visual tracking", in *Machine Vision and Applications*, vol. 14, No. 1, pp 50–58, April, 2003.
- [102] Stewart, C. V., "Robust Parameter Estimation in Computer Vision", in *SIAM Review*, vol. 41, No. 3, pp. 513-537, 1999.
- [103] Thrun, S., "Learning metric-topological maps for indoor mobile robot navigation", in *Artificial Intelligence*, vol. 99, No. 1, pp. 21-71, 1998.
- [104] Tissainayagam, P. and D. Suter, "Performance Measures for Assessing Contour Trackers", in *Int. Journal of Image and Graphics*, Vol. 2, No 2, pp. 343-359, April, 2002.
- [105] Toyama K. and G. Hager, "Incremental Focus of Attention for Robust Vision-Based Tracking", in *International Journal of Computer Vision*, vol. 35, No. 1, pp. 45-63, 1999.
- [106] Trahanias P. E., S. Velissaris and S. C. Orphanoudakis, "Visual Recognition Landmarks for Topological Navigation", in *Autonomous Robots*, Kluwer Acad., vol. 7, pp. 143-158, 1999.

- [107] Ukita N., and T.Matsuyama, “Real-Time Cooperative Multi-Target Tracking by Communicating Active Vision Agents”, in *IPSJ Transactions on Computer Vision and Image Media*, vol. 43, No. SIG-11(CVIM 5), pp. 64-79, 2002.
- [108] Vandapel N., Perception et sélection d'amers en environnement naturel pour la navigation d'un robot mobile, Thèse de doctorat de l'INPT, 2000.
- [109] Vincze, M., “Robust tracking of ellipses at frame rate”, in *Pattern Recognition*, vol. 34, No. 2, pp. 487-498, 2001.
- [110] Victorino, A.C., P. Rives and J.J. Borrelly, “A Relative Motion Estimation by Combining Laser Measurement and Sensor Based Control”, in Proc of *ICRA 2002*, pp. 3924-3929, 2002.
- [111] Winters, N. and J. Santos-Victor, “Information Sampling for vision-based robot navigation”, in *Robotics and Autonomous Systems*, vol. 41, pp. 145-159, 2002.
- [112] Weidner R.T. and R.L. Sells, *Elementary Classical Physics*, Allyn and Bacon, USA, Vol. 2, pp. 692-696, 1971.
- [113] Wu Y., And T. S. Huang, “A Co-inference Approach to Robust Visual Tracking”, in Proc. of IEEE *Int'l Conf. on Computer Vision (ICCV'2001)*, pp. 26-33, Vancouver, Canada, 2001
- [114] Xiang, T., S. Gong and D. Parkinson, “Autonomous Visual Events Detection and Classification without Explicit Object-Centred Segmentation and Tracking”, in *Proceeding BMVC 2002*, vol. 1, pp. 233-242, Cardiff, U.K., 2002.
- [115] Yoshida, T., A. Ohya and S. Yuta, “Autonomous Mobile Robot Navigation Using Braille Blocks in Outdoor Environment”, in Proc. of the *2nd International Symposium on Robotics and Automation ISRA'2000*, pp. 575-580, Monterey, Mexico, November, 2000.

Vision Dynamique pour la navigation d'un robot mobile.

Les travaux présentés dans cette thèse concernent l'étude des fonctionnalités visuelles sur des scènes dynamiques et ses applications à la robotique mobile. Ces fonctionnalités visuelles traitent plus précisément du suivi visuel d'objets dans des séquences d'images. Quatre méthodes de suivi visuel ont été étudiées, dont trois ont été développées spécifiquement dans le cadre de cette thèse. Ces méthodes sont : (1) le suivi de contours par un *snake*, avec deux variantes permettant son application à des séquences d'images couleur ou la prise en compte de contraintes sur la forme de l'objet suivi, (2) le suivi de régions par différences de motifs, (3) le suivi de contours par corrélation 1D, et enfin (4) la méthode de suivi d'un ensemble de points, fondée sur la distance de Hausdorff, développée lors d'une thèse précédente. Ces méthodes ont été analysées pour différentes tâches relatives à la navigation d'un robot mobile; une comparaison dans différents contextes a été effectuée, donnant lieu à une caractérisation des cibles et des conditions pour lesquelles chaque méthode donne de bons résultats. Les résultats de cette analyse sont pris en compte dans un module de planification perceptuelle, qui détermine quels objets (amers plans) le robot doit suivre pour se guider le long d'une trajectoire. Afin de contrôler l'exécution d'un tel plan perceptuel, plusieurs protocoles de collaboration ou d'enchaînement entre méthodes de suivi visuel ont été proposés. Finalement, ces méthodes, ainsi qu'un module de contrôle d'une caméra active (site, azimut, zoom), ont été intégrées sur un robot. Trois expérimentations ont été effectuées: a) le suivi de route en milieu extérieur, b) le suivi de primitives pour la navigation visuelle en milieu intérieur, et c) le suivi d'amers plans pour la navigation fondée sur la localisation explicite du robot.

Mots clefs : robotique mobile, navigation visuelle, vision dynamique, vision active, suivi visuel.

Dynamic vision for mobile robot navigation.

The work presented on this thesis concerns the study of visual functionalities over dynamic scenes and their applications to mobile robotics. These visual functionalities consist on visual tracking of objects on image sequences. Four methods of visual tracking has been studied, from which tree of them has been developed specifically for the context of this thesis. These methods are: (1) snakes contours tracking, with two variants, the former, to be able to applying it to a sequence of color images and the latter to consider form constraints of the followed object, (2) the tracking of regions by templates differences, (3) contour tracking by 1D correlation, and (4) the tracking method of a set of points, based on Hausdorff distance, developed on a previous thesis. These methods have been analyzed for different tasks, relatives to mobile robot's navigation. A comparison for different contexts has been done, given to a characterization of objects and conditions for which each method gives the best results. Results from this analysis has been take into account on a perceptual planification module, that determines which objects (plane landmarks) must be tracked by the robot, to drive it over a trajectory. In order to control the execution of perceptual plan, a lot of collaboration or chaining protocols have been proposed between methods. Finally, these methods and a control module of an active camera (pan, tilt, zoom), has been integrated on a robot. Three experiments have been done: a) road tracking over natural environments, b) primitives tracking for visual navigation over human environments and c) landmark tracking for navigation based on explicit localization of robot.

Keywords: mobile robotics, visual navigation, dynamic vision, active vision, visual tracking