Techniques visuelles pour la détection et le suivi d'objets 2D

Rafiq Sekkal

Thèse

THESE INSA Rennes
sous le sceau de l'Université européenne de Bretagne
pour obtenir le titre de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Traitement du signal et de l'image

présentée par
Rafiq SEKKAL

ECOLE DOCTORALE : *MATISSE*
LABORATOIRE : *IRISA – UMPR6047*

Techniques visuelles pour la détection et le suivi d'objets 2D

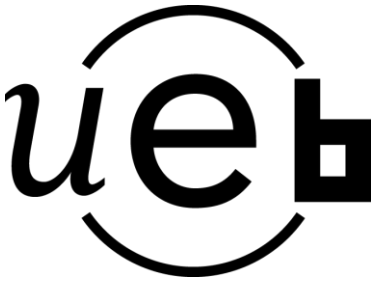
Muriel PRESSIGOUT
Maître de conférences à l'INSA de Rennes / Invitée

Thèse soutenue le 28/02/2014
devant le jury composé de :

Joseph RONSIN
Professeur à INSA de Rennes / Président
Luc BRUN
Professeur à ENSI de Caen / Rapporteur
Vincent FREMONT
Maître de conférences HDR UT de Compiègne / Rapporteur
Ferran MARQUES
Professeur à UPC Barcelone / Examinateur
Marie BABEL
Maître de conférences HDR à l'INSA de Rennes / Directrice de thèse

Techniques visuelles pour la détection et le suivi d'objets 2D

Rafiq SEKKAL



En partenariat avec

"Computer science is no more about computers
than astronomy is about telescopes"
Edsger Dijkstra

A mes parents, ma famille
A ma femme...

Remerciements

Cette thèse est le fruit de trois années de recherche effectuées au sein de l'IRISA. Je tiens à remercier toutes les personnes qui ont participé, de près ou de loin, à la réalisation de ces travaux.

Je tiens tout d'abord à remercier spécialement ma directrice de thèse, Mme Marie Babel pour la qualité de son encadrement, sa disponibilité permanente, ses encouragements et ses conseils pertinents tout au long de cette thèse.

Je remercie Mr Joseph RONSIN pour m'avoir fait l'honneur de présider mon jury de thèse ainsi que les rapporteurs M. Luc BRUN et Mr Vincent FREMONT d'avoir accepté de prendre le temps d'évaluer cette thèse à travers leurs remarques pertinentes qu'ils ont soulevées. Je remercie également Mme Muriel PRESSIGOUT et Mr Ferran MARQUES pour avoir participé à mon jury de thèse.

J'adresse aussi mes remerciements à Mr Ferran Marques Pour son accueil chaleureux tout au long de mon séjour au sein de son laboratoire à Barcelone.

Je remercie aussi l'ensemble de l'équipe Lagadic en commençant par le chef d'équipe François CHAUMETTE pour m'avoir accepté dans l'équipe. Un merci particulier à François PASTEAU pour toute son aide, ses conseils pertinents durant ses trois dernières années. Un grand merci à Aurélien, Bertrand, Antoine, Clément et Céline avec qui les journées au laboratoire étaient pleines de travail, de sérieux et aussi d'humour.

Un immense "Merci" particulier à mes très chers parents, pour leur amour infini, leur présence, leurs prières et leur grand soutien durant toute ma vie ainsi que mon très cher frère Fouad, et mes soeurs, Hadjira et Zoubida. Un grand merci aussi à Nawel et Fayçal qui m'ont soutenu et qui me soutiennent toujours depuis mon arrivée en France. Sans oublier Ahmed et Lila et mes petits bouts de choux Ilham, Younes, Racim, Ayoub et Hiba.

Mes chaleureux remerciements à ma femme et bien-aimée Zahira pour m'avoir soutenu, encouragé et rassuré dans les moments difficiles de la vie d'un doctorant ainsi que ma belle-famille pour leur sympathie et leur encouragement.

Pour Finir, un grand merci à tous mes amis Hakim, Adel, Hicham, Mohammed, Djawida pour avoir partagé cette aventure en France.

Table des matières

Introduction générale

La motivation première des travaux présentés dans cette thèse est de développer des techniques visuelles appropriées pour l'aide à la navigation des personnes à mobilité réduite utilisant des fauteuils électriques.

Ces travaux de thèse s'inscrivent donc en partie dans un projet collaboratif appelé APASH (Assistance au Pilotage pour l'Autonomie et la Sécurité des personnes Handicapées) pour lequel l'équipe Lagadic est partie prenante. Cette assistance peut s'envisager sous différentes formes :

- franchissement de portes,
- utilisation d'un ascenseur,
- évitement d'obstacles,
- navigation dans un couloir.

Figure 1 – Fauteuil 6-roues de série robotisé

La figure 1 montre le fauteuil utilisé dans notre projet. Comme pour tout fauteuil 6 roues, le rayon de giration est très court car le fauteuil tourne sur son propre axe. La manipulation est donc plus intuitive pour l'utilisateur.

Lorsqu'une personne handicapée utilise un fauteuil à longueur de journée, que ce soit à domicile, au travail ou à l'extérieur, la tâche de navigation devient de plus en plus difficile. La fatigue peut être visuelle, à force de se concentrer sur l'objectif à atteindre et la tâche à effectuer. La fatigue peut être aussi musculaire à force de manier le joystick pour guider le fauteuil lors de la navigation. Le projet APASH consiste donc à fournir une assistance à la navigation et non pas une navigation autonome du fauteuil roulant. Ceci se traduit par la correction de la trajectoire lorsque cela est possible.

Parmi les tâches que doit accomplir le projet APASH, le franchissement sûr de portes est une priorité. Franchir une porte n'est pas une tâche évidente, le risque de collision avec les montants de portes est très élevé du fait de l'empattement du fauteuil et des habitudes de navigation de la personne.

Afin de proposer une assistance au franchissement de porte, le système doit donc être capable de détecter les portes dans l'environnement où il évolue. La détection des portes sera ensuite considérée comme une initialisation pour le module de navigation afin de planifier la trajectoire du fauteuil.

Par ailleurs, le projet APASH est un projet en collaboration avec deux industriels. *AdvanSEE* est une entreprise spécialisée dans la fabrication des cartes embarquées et *Ergovie* est une entreprise spécialisée dans la personnalisation des fauteuils pour les personnes handicapées. La solution issue de ce projet doit donc être donc capable d'intégrer les fauteuils roulants du marché et ainsi de pouvoir être proposée en option pour les futurs possesseurs de fauteuil. Le module doit donc être de faible coût afin que sa mise en oeuvre soit envisageable.

Nous avons réfléchi au type d'équipement que l'on peut utiliser pour cette problématique. Pour cela, nous avons opté pour une solution basée vision. Nous avons donc équipé notre fauteuil par une caméra monoculaire couleur. Au contraire d'un dispositif laser classiquement utilisé à des fins de navigation, cette caméra peut être achetée à un prix très raisonnable par rapport au prix du fauteuil.

Afin de pouvoir faire la navigation visuelle, le système doit être en mesure de reconnaître l'environnement qui l'entoure. Dans notre cas, le fauteuil évolue dans un environnement intérieur de type hôpital, bureau ou encore dans le cadre d'un maintien à domicile. Ce type d'environnement est caractérisé par ses propriétés structurelles dues à l'intervention de l'homme : murs, sol, plafond, fenêtres, couloirs, portes... Ces éléments vont induire un ensemble de primitives élémentaires de type lignes, points de fuite, formes géométriques (cercles,

rectangles) et de la symétrie. Ces caractéristiques seront exploitées comme des a priori pour la détection et la localisation des repères visuels.

Une des caractéristiques du projet APASH est qu'on ne dispose pas de carte de l'environnement. Pour faire de la navigation dans un milieu inconnu, il faut donc être capable de détecter des repères afin de pouvoir se localiser. Pour cela, il faudra déterminer quels amers visuels sont les plus pertinents. Dans le cas de la navigation en intérieur, on peut considérer plusieurs types d'amers, comme les portes, les fin/début de murs...

La première contribution de cette thèse consiste ainsi en une technique de détection et suivi de portes basée vision. Notre solution a été conçue dans un premier temps pour une détection et un suivi à partir d'un couloir. Le traitement est en effet spécifique pour chaque type d'environnement (couloir, bureau, chambre), du fait des caractéristiques géométriques propres à chaque environnement (par exemple dans un couloir, les murs sont parallèles et forment une symétrie dans la perspective de la scène).

La détection des portes servira alors d'initialisation à une technique de suivi de portes proposée comme une seconde contribution dans cette thèse. Le suivi de portes permet de fournir des informations en temps réel de la localisation de la porte dans l'image. Ce suivi est d'une importance primordiale pour la tâche de navigation qui va utiliser les portes comme repère visuel.

Par ailleurs, naviguer de façon sûre, c'est s'assurer que le fauteuil ne court aucun risque de collision. Dans un environnement intérieur, rouler avec un fauteuil roulant comporte des risques de collision, soit avec la structure du bâtiment (mur, portes), soit avec des éléments dynamiques de la scène (personnes, brancard, fauteuil roulant). La détection d'obstacles demeure alors une tâche nécessaire.

Les obstacles sont de tailles, formes et couleurs inconnues du système de détection. Il est donc impossible de les modéliser pour les détecter. De plus, leur trajectoire n'est pas toujours uniforme : les objets complexes impliquent une trajectoire complexe. La détection de ce type d'objets n'est pas une tâche facile car en plus de leur mouvement complexe, il faut ajouter le mouvement de la caméra embarquée sur le fauteuil.

Pour détecter et suivre les objets inconnus, nous avons fait donc appel aux techniques d'analyse d'images basées sur la segmentation. Notre troisième contribution consiste ainsi en un algorithme de segmentation d'images fixes.

Cet algorithme est capable de fournir une représentation pseudo-sémantique de la scène. Les régions sont extraites en fonction de leurs similarités. Cette technique est caractérisée par des propriétés de scalabilité, et nous proposons de combiner dans le même algorithme la scalabilité spatiale et la scalabilité sémantique. La hiérarchie obtenue permet ainsi de fournir des résultats à différents niveaux de sémantique. D'autre part, nous allons montrer que l'utilisation de la multirésolution permet de réduire le coût de calcul de la segmentation.

La dernière contribution dans cette thèse consiste en une extension de notre algorithme de segmentation d'images en vue de la segmentation spatio-temporelle des séquences d'images. Cette technique permet de suivre les régions d'une image à une autre. Le véritable défi de cette méthode est d'assurer la cohérence spatio-temporelle des régions du moment de leur apparition jusqu'à leur disparition.

Les solutions proposées dans cette thèse seront utilisées pour la navigation d'un fauteuil roulant électrique avec une personne handicapée à bord. Modifier un tel dispositif requiert une grande prudence dans la solution embarquée. En effet, la responsabilité du concepteur d'un tel dispositif est engagée dans le cas d'un accident causé par le module. L'erreur n'est donc pas envisageable au risque de mettre la vie d'autrui en danger. Pour cela, les solutions proposées dans cette thèse et ceux qui vont les compléter doivent faire l'objet d'une vérification ultérieure minutieuse afin de repérer tout défaillance potentielle.

Ce manuscrit se compose de quatre chapitres comme présentés dans la figure 2.

- **Chapitre 1** : Dans le premier chapitre, nous allons présenter une nouvelle technique de détection des portes dans un environnement de type couloir. Cette technique se base sur des descripteurs géométriques capables d'identifier les portes. L'originalité de ces travaux est l'utilisation de descripteurs purement visuels pour la détection de porte. Ainsi, nous montrons la possibilité de détecter des portes quel que soit leur état (ouvertes/fermées), leur couleur, et même leur distance à la caméra.

- **Chapitre 2** : Le second chapitre présente la technique de suivi de porte proposée dans cette thèse. Cette technique a été inspirée des solutions de suivi de d'objets existant dans la littérature et a été adaptée à notre type d'objet à suivre. Nous avons exploité pour cela l'a priori du mouvement du modèle ainsi qu'une représentation simplifiée de la porte pour améliorer son suivi.

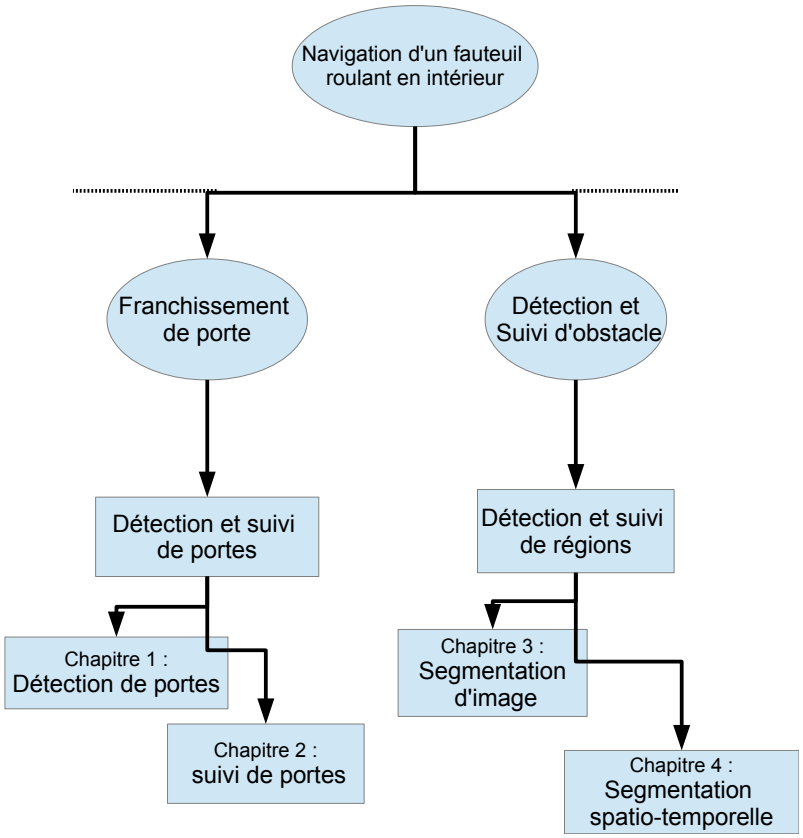


Figure 2 – Organisation globale du manuscrit de thèse

– **Chapitre 3** : Le troisième chapitre introduit la détection d'objets inconnus dans la scène. Ceci est rendu possible à travers une technique de segmentation d'image capable de fournir des représentations pseudo-sémantiques de la scène observée. L'originalité de ces travaux consiste en la fusion des propriétés de scalabilité spatiale et sémantique afin de proposer la meilleure représentation possible.

– **Chapitre 4** : Le dernier chapitre est une continuité des travaux de segmentation, et introduit à savoir une nouvelle technique de segmentation spatio-temporelle basée sur la projection des contours des régions. Cette technique permettra de suivre les régions quelle que soit leur forme et quelle que soit leur trajectoire.

Chapitre 1

Détection des portes dans un couloir

Dans ce chapitre, nous allons traiter de la problématique de détection des portes à des fins de navigation dans un milieu intérieur. La solution proposée devra offrir à l'utilisateur la possibilité de réaliser des tâches en toute sécurité comme :

- la navigation dans un couloir,
- le franchissement de portes.

Dans un contexte de navigation visuelle, il est indispensable d'avoir une vue d'ensemble sur l'environnement qui nous entoure pour pouvoir y évoluer et circuler. Toute information est bonne à prendre : la structure de l'environnement, les repères à suivre, les obstacles à éviter, etc. Ces informations vont servir à se localiser dans l'environnement et accomplir les tâches souhaitées.

Afin de se localiser, il est nécessaire de se baser sur des repères visuels sémantiques stables. La détection de ces repères est une problématique qui a suscité beaucoup d'intérêt, et a fait l'objet de nombreux travaux à la fois dans le domaine de la navigation et la reconstruction de l'environnement.

Différents types de repère peuvent être utilisés. De façon non exhaustive on peut citer :

- **point de fuite** : un repère des plus basiques permettant de déterminer l'orientation de la perspective ;
- **points saillants** : de type Haar, SIFT etc. ces points possèdent des caractéristiques bien spécifiques permettant de les identifier et de les suivre lors

de la navigation ;
- **amers visuels** : ces repères sont créés spécialement pour la localisation (typiquement un cône posé par terre). Leurs caractéristiques sont bien modélisées et leur positions permettent une localisation immédiate ;
- **Objets de l'environnement** : ce dernier type de repère peut être représenté par n'importe quel objet faisant partie intégrante de l'environnement (porte, fenêtre, escalier, etc.).

Dans la dernière catégorie, le type d'objet à détecter est lié à l'application pour laquelle le système est développé. Dans notre cas, afin d'offrir la possibilité à l'utilisateur de pouvoir franchir des portes, il indispensable de connaître leur position relative au fauteuil afin de réaliser la bonne trajectoire.

Les portes sont des structures sémantiques stables pour la localisation dans un environnement intérieur c'est pourquoi il est indispensable de les détecter . La détection permet d'initialiser à la fois l'algorithme de localisation et l'algorithme de suivi (*tracking*) de porte que l'on va détailler dans le chapitre 2 pour assurer la mise à jour en temps réel de la position du fauteuil. L'enjeu est de taille : une bonne détection de porte permet une bonne localisation et assure la sécurité lors du franchissement. Pour cela, il faut prendre en compte plusieurs aspects : le modèle de la porte, la structure du couloir.

Dans ce chapitre, nous allons présenter un algorithme de détection des portes dans un environnement intérieur de type couloir. Dans la section 1.1, nous présentons un ensemble de techniques de détection de portes. Ensuite, nous présentons le schéma général ainsi que les descripteurs utilisés dans les sections 1.2 et 1.3. Puis, la section 1.4 aborde une technique de détection des zones sol/mur dans une perspective de couloir. Ensuite, la localisation du fauteuil dans le couloir est présentée dans la section 1.5. La technique de reconnaissance des portes est détaillée dans la section 1.6. Enfin dans la section 1.7, nous présentons les expériences et les résultats effectués pour la validation de notre algorithme.

1.1 État de l'art : Détection de portes

De nos jours, il existe peu de travaux sur la seule détection des portes. Ceci peut être dû à la spécificité de la problématique qui est liée à son application. Les techniques de détection de portes basées vision se différencient par les descripteurs utilisés (cf. figure 1.1). Les portes sont généralement caractérisées par leur forme rectangulaire [Stoeter 00, Chen 08, Tian 10, Kim 11, Shi 06] qui permet de démarquer la surface de la porte par rapport à celle des murs. D'autres travaux

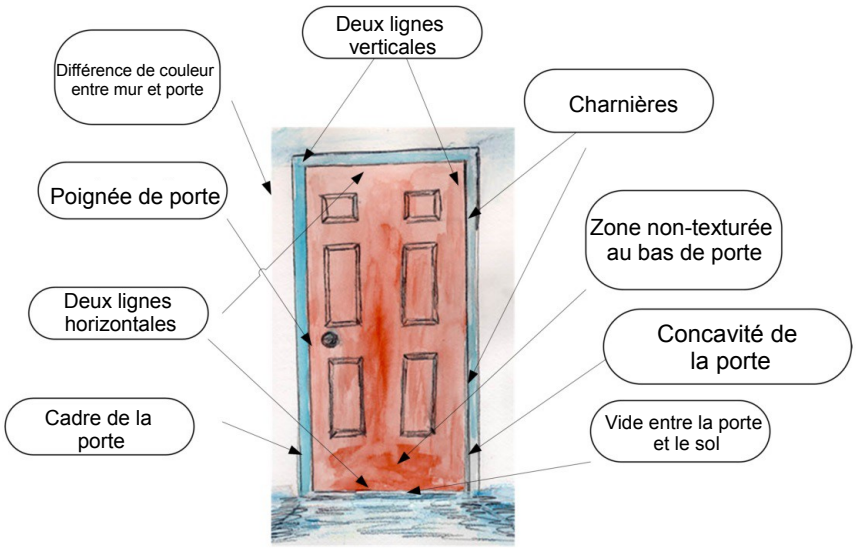


Figure 1.1 – Representation des descrpiteurs d'une porte [Hensler 10]

considèrent les coins de portes comme descripteurs pour la détection de la forme de porte [Murillo 08, Yang 10, Tian 10]. La couleur est aussi utilisée pour décrire une porte qui se trouve généralement de couleur uniforme et différente de celle des murs [Chen 08, Murillo 08]. Certains auteurs n'hésitent pas à combiner plusieurs descripteurs pour renforcer leur technique [Chen 08, Murillo 08].

[Kim 94] propose une approche générique pour la reconnaissance des objets. Il considère lui aussi la porte comme une forme $\sqcap$. Cette forme est définie par les deux montants et le haut de la porte. Sa technique consiste à chercher des formes $\sqcap$ respectant le critère hauteur/largeur.
Dans [Stoeter 00], les portes sont identifiées par leur montants. Les contours de l'image sont renforcés en appliquant une opération de dilatation morphologique suivie d'une érosion. Ensuite, les lignes verticales sont extraites. Les portes sont localisées par rapport aux dimensions des lignes verticales et celles du couloir. Cependant, la détection de portes ne prend pas en compte les changements affectés par la perspective.

Une technique de détection de porte à base de logique floue a été proposée par [MuñozSalinas 04]. Cette technique utilise la transformée de Hough pour extraire les segments dans l'image. Les relations entre les segments sont ensuite analysées en utilisant de la logique floue. Différents descripteurs sur les lignes sont alors définis (taille, direction et distance entre les lignes). Une classification des lignes est effectuée pour définir les lignes verticales et horizontales en fonction de leur direction. Ensuite, à travers des règles de décision, les portes sont détectées en

fonction de l'identification à la fois d'une ligne horizontale correspondant au haut de la porte et de deux lignes verticales correspondant aux montants de la porte.

L'ajout des informations de l'environnement permet d'améliorer la détection. Ainsi, [Shi 06] utilise une description de la structure du couloir. La structure globale du couloir est détectée en fonction des lignes du couloir. Ensuite, toute forme $\sqcap$ est considérée comme une porte si ses extrémités coïncident avec les lignes du couloir. Dans cette technique, la détection des portes est conditionnée par la détection de la forme $\sqcap$, notamment la ligne qui correspond au haut de la porte. Cependant, dans une image de perspective, la ligne de haut de porte peut être d'une petite taille rendant sa détection parfois impossible. Par conséquent, la détection de la forme ainsi que de la porte sera vouée à l'échec. De plus, aucune vérification sur les proportions de la forme $\sqcap$ détectée n'est effectuée. Ceci peut causer des mauvaises détections dans le cas où on a des objets sur le mur en contact avec le sol (radiateurs, meubles ...).

[Chen 08] présente une approche supervisée : un ensemble de descripteurs est utilisé pour la reconnaissance des portes (couleurs, texture, contours). De plus, un descripteur spécifique du bas de porte a été défini : il correspond au vide se trouvant entre le bas de la porte et le sol. Ce vide est caractérisé généralement par une couleur sombre lorsque la porte est fermée. La détection est assuré par un algorithme de type Adaboost permettant de combiner un ensemble de classifieurs pour chaque descripteur cité précédemment. Cependant, la détection des portes est limitée aux portes qui sont fermées. De plus, pour pouvoir détecter une porte, la caméra doit se placer face à la porte à une distance très limitée, ce qui rend son utilisation impossible pour des applications qui nécessitent une distance importante pour la planification de la trajectoire par exemple.

Murillo [Murillo 08] propose quant à lui une approche probabiliste pour la détection des portes. A partir d'un ensemble d'apprentissage, les descripteurs visuels de portes sont extraits en vue de reconnaître la couleur et la forme de la porte. Ensuite un ensemble d'hypothèses est généré. En particulier, le ratio/hauteur largeur des portes (la forme) doit être respecté. Ces hypothèses sont ensuite évaluées par un maximum de vraisemblance à partir des modèles construits lors de la phase d'apprentissage des descripteurs visuels (la couleur). Cette méthode donne de bons résultats pour les portes de couleurs différentes des murs. Cependant, lorsque les portes sont de la même couleur que les murs, leur détection devient alors impossible. De plus, lorsque les portes sont ouvertes, la couleur dépend alors de l'espace qui se trouve derrière la porte, ce qui peut causer la non détection de la porte.

Une solution de détection de portes pour assister les personnes non-voyantes dans des environnements inconnus a été proposé par [Tian 10]. Les auteurs combinent à la fois les coins et les contours pour détecter des portes. Tout d'abord, les coins sont détectés à partir de l'image de contours (filtre de Canny). Ensuite, les lignes de la porte sont estimées en fonction de la correspondance entre les contours obtenus, et la ligne qui relie deux coins estimés. Toutefois, l'angle de la porte dans une vue en perspective est fixé empiriquement dans un intervalle sans prendre en compte la position et la rotation de la caméra.

Il existe aussi d'autres techniques de détection de portes qui font appel à d'autres types de capteur que nous n'allons pas présenter dans ce manuscrit (laser, radar, sonar ...)[ElKaissi 07, Carinena 04]. D'autres techniques appelées hybrides combinent des approches basées vision et un autre capteur pour améliorer la détection des portes [Hensler 10, Anguelov 04]. Bien que ces capteurs peuvent donner des informations précises sur l'environnement (profondeur), leur coût de réalisation reste toutefois le principal frein pour leur application dans notre contexte.

Dans notre approche, nous allons proposer une approche de détection de portes dans un environnement de type couloir. Pour cela, nous allons devoir tenir compte de la structure globale d'un couloir. Il s'agit donc dans un premier temps de procéder à l'identification du sol et des murs afin de considérer les murs comme des zones de recherche pouvant contenir potentiellement des portes. De plus, il faut déterminer la position de la caméra dans le couloir qui joue un rôle très important dans la forme de la porte dans l'image. Pour cela, nous allons devoir utiliser quelques descripteurs visuels qui seront décrits dans la section suivante.

1.2 Schéma général de la détection de portes

Dans notre schéma, la détection des portes dans un environnement intérieur de type couloir consiste à identifier les formes rectangulaires se trouvant sur les murs. Ces formes devront respecter des proportions correspondant aux normes standard des portes relatives à l'accessibilité aux personnes à mobilité réduite. Les portes se trouvant dans le même couloir partagent généralement les mêmes caractéristiques (couleur, hauteur, largeur).

Le défi est alors d'estimer les dimensions des portes dans le couloir en fonction des descripteurs géométriques. A partir d'une image fournie par une caméra bien calibrée, la forme rectangulaire de la porte sera conservée et composée de droites bien définies représentant les montants et le haut/bas des portes.

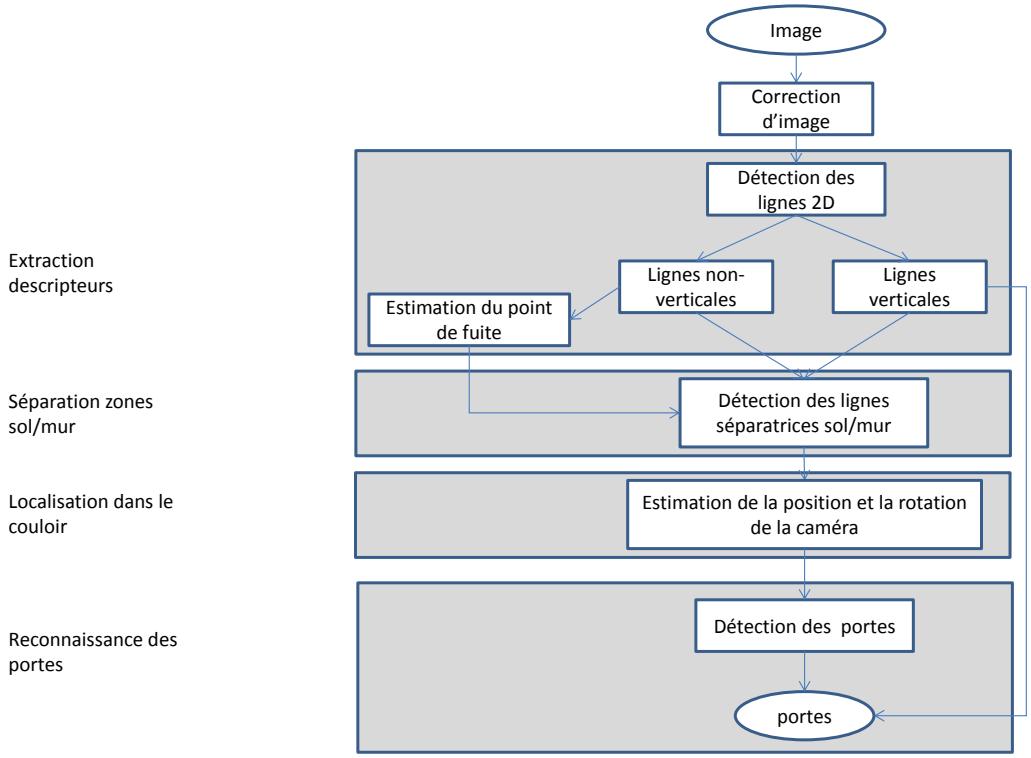
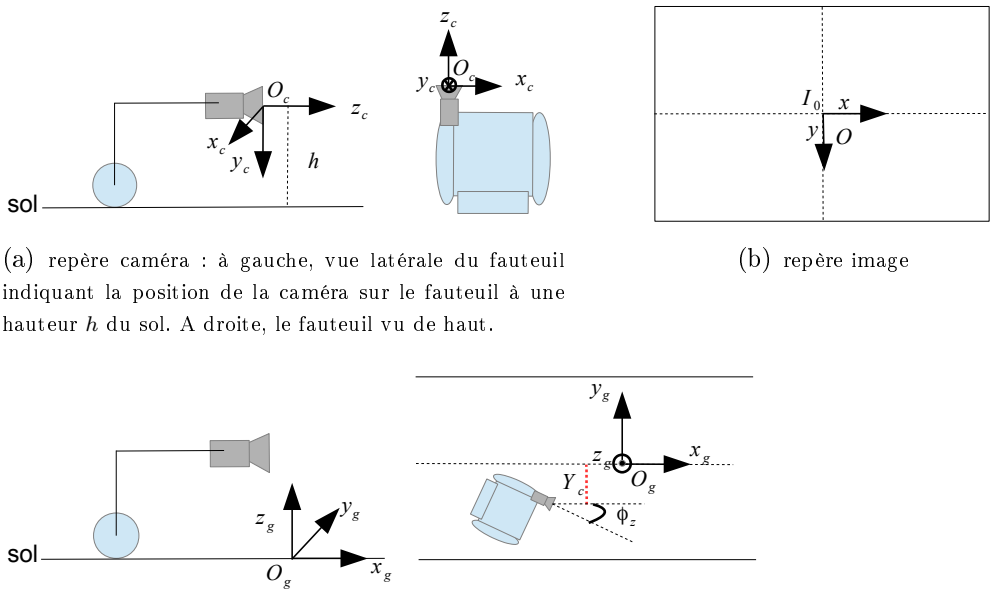


Figure 1.2 – Le schéma général de la détection de portes

Le schéma général de la détection de portes est représenté dans la figure 1.2. L'algorithme se compose de quatre étapes successives :

1. **Extraction des descripteurs :** dans un premier temps, l'ensemble des descripteurs sont extraits, notamment les lignes nécessaires à la détection de porte. Les lignes vont permettre d'identifier les montants de la porte et d'estimer les points de fuite. Cette étape est détaillée dans la section 1.3
2. **Détection sol/mur** : afin d'assurer une bonne détection des portes, nous localisons les zones correspondant aux murs dans l'image. Cette recherche repose sur la position du point de fuite, les lignes détectées et les a priori dont nous disposons sur la structure géométrique d'un environnement de type couloir. Cette technique est présentée en détails dans la section 1.4
3. **Localisation dans le couloir** : la forme de la porte subit des transformations liées à la position et la rotation de la caméra dans le couloir. Ainsi, nous proposons une technique d'estimation de la position relative de la caméra ainsi que sa rotation dans le couloir (1.5). Cette technique est basée



(a) repère caméra : à gauche, vue latérale du fauteuil indiquant la position de la caméra sur le fauteuil à une hauteur h du sol. A droite, le fauteuil vu de haut.

(b) repère image

(c) repère global : à gauche, vue latérale et à droite vue de haut indiquant la position Y_c de la caméra ainsi que sa rotation ϕ_z

Figure 1.3 – Définition des repères

sur les résultats de détection du sol/mur ainsi que les connaissances a priori de la scène.

4. **Reconnaissance des portes** : enfin, la reconnaissance des porte est effectuée en fonction de la détection des formes rectangulaires sur les zones mur. La décision de détection dépend de sa taille, de sa profondeur et de l'orientation de la caméra. La section 1.6 présente le processus de reconnaissance des portes.

Notations

Dans cette section nous introduisons les notations utilisées dans ce chapitre. Les repères sont illustrés dans par figure 1.3. Nous notons

- $O_g(x_g, y_g, z_g)$ le repère global du couloir,
- $O_c(x_c, y_c, z_c)$ le repère caméra,
- $O(x, y)$ le repère du plan de l'image,
- (u_0, v_0, p_x, p_y) les paramètres intrinsèques de la caméra avec u_0 et v_0 les coordonnées de la projection du centre optique de la caméra sur le plan image, p_x et p_y les distances focales suivant x et y exprimées en largeur et en hauteur de pixels,

- $(X_c, Y_c, Z_c, \phi_x, \phi_y, \phi_z)$ paramètres extrinsèques de la caméra exprimés dans le repère global du couloir,
- $I_0(u_0, v_0)$ le point principal dans l'image,
- $vp(x_{vp}, y_{vp})$ un point de fuite dans le repère image.

1.3 Descripteurs géométriques

La détection de portes repose sur un ensemble de descripteurs calculés dans une première phase. Comme nous l'avons montré dans la section 1.1, nombreux sont les descripteurs visuels utilisés dans l'état de l'art. Nous allons nous intéresser ici à la détection des lignes. Ces descripteurs sont naturellement utilisés pour représenter des formes rectangulaires. Dans la suite, nous allons présenter l'outil de détection de lignes utilisé dans notre technique.

1.3.1 Extraction des lignes

Il existe différentes approches de détection des segments. La plus répandue consiste en la transformée de Hough [Duda 72]. Elle est notamment utilisée dans [Oliver 06] pour la détection des portes. Cette technique consiste à estimer les paramètres de la ligne en appliquant une transformée vers l'espace Hough. Cependant, durant nos expérimentations, nous avons remarqué que cette technique est très sensible aux paramètres d'extraction. Par exemple, une fenêtre très petite peut causer la décomposition d'une ligne en plusieurs petit segments. Si, par contre, on augmente la taille de fenêtre, on obtient des lignes incohérentes issues de fusions de segments qui ne sont pas forcément connexes. De plus, cette technique est plus efficace pour la détection des points de fuite plutôt que l'extraction de lignes précises.

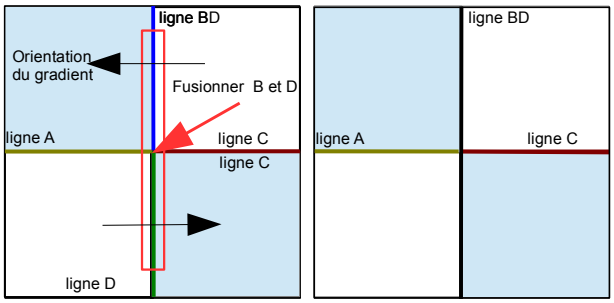
Pour la détection des lignes, nous proposons d'utiliser un algorithme de détection de segments LSD (*Line Segment Detection*) détaillé dans l'article [vonGioi 12]. Le LSD est utilisé pour détecter les contours formant des droites dans les images. L'algorithme commence par estimer l'angle du gradient à chaque pixel. Ensuite, les pixels partageant le même angle de gradient sont fusionnés en régions à travers un algorithme de croissance de régions. Les régions correspondant à des lignes sont approximées par des formes rectangulaires définies par une orientation. Ensuite, un score est attribué à chaque rectangle en fonction du nombre de points alignés (les pixels ayant la même angle de gradient que l'orientation du rectangle). Si le score est supérieur à un certain seuil prédéfini, alors le rectangle est considéré comme un segment.

1.3.2 Classification et fusion des lignes

Dans une perspective de couloir, on constate que les lignes peuvent être classées en deux groupes :

- **Lignes verticales :** ce sont les lignes qui correspondent aux montants des portes et les éléments se trouvant sur les murs (cadres, placards, etc.). Ces lignes sont dites verticales si leur angle θ est compris entre $\frac{\pi}{2} - \alpha < \theta < \frac{\pi}{2} + \alpha$ avec α un paramètre permettant de moduler l'intervalle de décision.
- **Lignes non verticales :** le reste des lignes contribuent au calcul des points de fuite visibles dans l'image.

Le LSD est basé sur un processus de fusion des points dans l'image qui partagent la même direction de gradient. Cependant, lorsque la direction du gradient change, l'algorithme le considère comme une nouvelle ligne même si celle-ci est connexe avec une autre. Par exemple, quand on considère l'image 1.4(a), le LSD extrait quatre lignes au lieu de deux. Ainsi, les montants des portes peuvent être définis en plusieurs segments suivant l'orientation du gradient qui peut changer pour plusieurs raisons (porte ouverte avec fenêtre donnant plus de lumière par exemple). Pour pallier ce problème, on applique une fusion des lignes verticales (cf. figure 1.4(b)).



(a) (b)

Figure 1.4 – Exemple de fusion de lignes : dans (a), quatre lignes sont détectées à cause d'un changement de gradient. (b) Lors de la fusion, les deux lignes B,D sont connexes par leur extrémité située au centre de l'image et partagent le même angle ($\frac{\pi}{2}$). Les deux lignes sont fusionnées et on obtient alors trois lignes A, C, et BD.

Les lignes fusionnées sont celles se trouvant sur la même droite. En effet, pour chaque ligne, on estime les paramètres de la lignes a, b et c dans le plan de l'image

en fonction de ses deux extrémités $ex1(x_{ex1}, y_{ex1})$ et $ex2(x_{ex2}, y_{ex2})$ selon

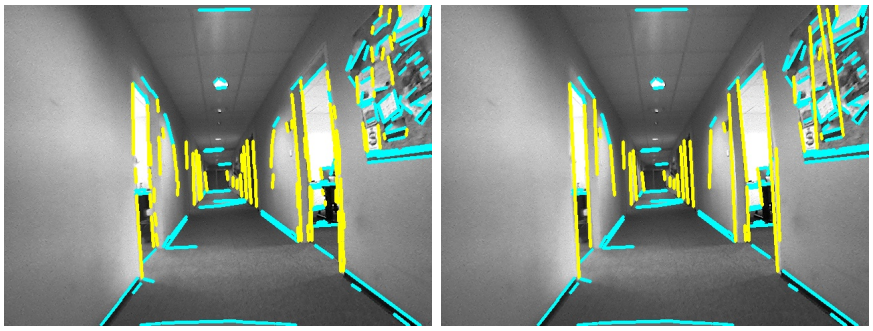
$$ax + by + c = 0 \tag{1.1}$$

Ensuite, pour chaque ligne verticale, nous estimons les distances D_{ex1} et D_{ex2} de ses extrémités $ex1$, $ex2$ avec la droite $ax + by + c = 0$ selon :

$$D_{exi} = \frac{ax_{exi} + by_{exi} + c}{\sqrt{a^2 + b^2}} \quad , i = \{1, 2\} \tag{1.2}$$

Ensuite, si D_{ex} est inférieur au seuil D_{th} fixé empiriquement alors les deux lignes sont fusionnées. Donc, si deux lignes verticales se trouvent sur la même droite, alors elle seront fusionnées.

La figure 1.5 présente les résultats de la fusion des lignes avec un $D_{th} = 5$ pixels. Les lignes verticales (en jaune) sont représentées avant (figure 1.5(a)) et après la fusion (figure 1.5(b)). On peut constater que dans la première image, les lignes verticales sont composées de plusieurs petits segments. Dans la porte à gauche, on remarque un changement de gradient qui se produit sur le montant gauche de la porte : en haut, du gris vers du blanc et en bas, du gris vers du noir. Ceci cause la détection de deux lignes distinctes. Après la fusion, on remarque que ces deux lignes sont fusionnées en une seule ligne verticale permettant de mieux représenter le montant de la porte.



(a) (b)

Figure 1.5 – Apport de la fusion des lignes verticales : à gauche, lignes sans fusion. A droite, les lignes sont fusionnées.

1.4 Localisation sol/mur

La deuxième étape consiste à localiser les zones correspondant au sol et aux murs afin de définir des zones de recherche des portes sur les murs. Afin de pou-

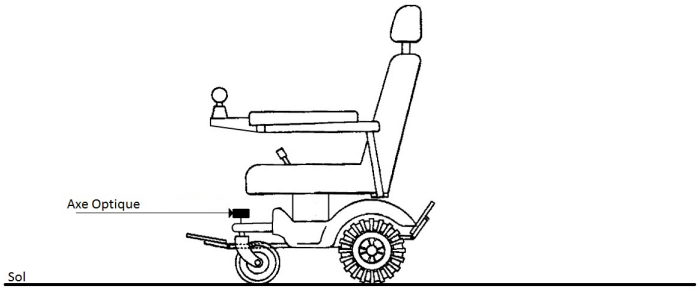


Figure 1.6 – Orientation de la caméra : l'axe optique est toujours horizontal et donc parallèle au sol.

voir réaliser la localisation sol/mur, il faut connaître au préalable la position du point de fuite visible dans l'image. Dans la section suivante, nous allons présenter la technique d'estimation du point fuite utilisée. Ensuite, nous présentons la technique de détection des zones sol/mur dans le couloir. Nous introduisons à la fin une technique de localisation de la caméra dans le couloir.

1.4.1 Calcul des points de fuite

Le point de fuite fournit des informations de la perspective de la scène. En effet, le point de fuite permet de regrouper toutes les lignes parallèles dans le monde réel. Si on souhaite reconstruire la structure globale du couloir (sol/murs), il est donc indispensable de connaître la position du point de fuite. De plus, dans une vue de perspective en profondeur, le point de fuite permet de déterminer l'orientation de la caméra et d'avoir une idée sur la structure globale de la scène.

L'estimation des points de fuite reste un problème ouvert, surtout lorsqu'on veut concevoir un système dont la précision et la faible complexité sont les critères à prendre en compte. Le calcul de points de fuite est une problématique qui a déjà été abordée un certain nombre de fois [Collins 89, Shufelt 99, Bazin 12]. Cependant, la méthode proposée par Rother [Rother 00] utilisant la projection sur une sphère gaussienne a fait ses preuves. Elle est notamment utilisée, moyennant quelques améliorations, dans le projet ATIP [Boulanger 06]. La méthode présentée dans cette section est très similaire, et a fait l'objet d'un stage d'étudiant de Master que j'ai encadré [Baptiste 13] en collaboration avec mon directeur de thèse.

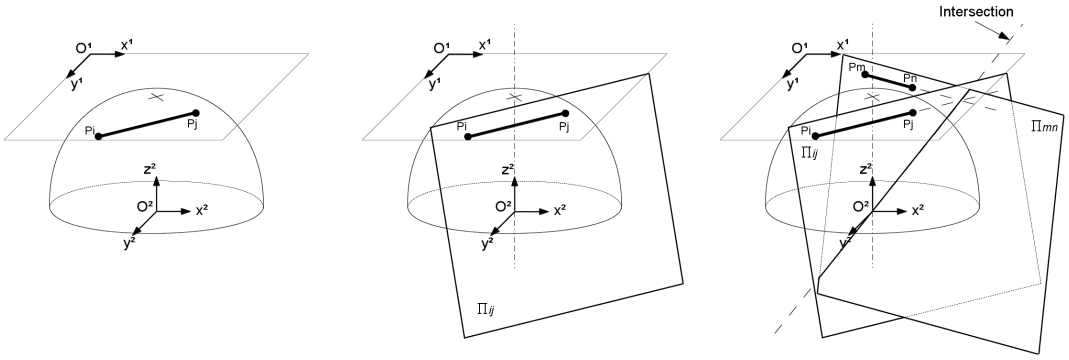
Les points de fuite sont calculés à partir des lignes non verticales. Comme l'orientation de la camera est parallèle au sol (cf. figure 1.6), les lignes verticales définissent un point de fuite infini. Ainsi, ne pas les considérer réduit considéra-

blement le nombre de lignes à prendre en compte, et le calcul des points de fuite devient alors bien plus rapide.

Pour pouvoir calculer les points de fuite finis et infinis, toutes les lignes non verticales sont projetées sur une demi-sphère de diamètre correspondant à la hauteur de l'image (figure 1.7(a)). Le sommet de la demi-sphère correspond au point principal.

Soient P_i et P_j deux points situés sur le plan de l'image $O^1(x_1, y_1)$ formant un segment détecté par LSD. On définit un plan Π_{ij} contenant ce segment et le centre de la sphère. Ce plan est représenté par son vecteur normal $\overrightarrow{N_{\Pi_{ij}}}$. Il est calculé via un produit vectoriel entre les deux points, en utilisant leurs coordonnées par rapport au repère O^2 situé au centre de la sphère (figure 1.7(b)), selon

$$\overrightarrow{N_{\Pi_{ij}}} = P_i \times P_j. \tag{1.3}$$



(a) Position de la sphère et des repères

(b) Plan Π_{ij} formé par les points P_i, P_j et O^2

(c) Intersection entre les plans Π_{ij} et Π_{mn}

Figure 1.7 – Technique de projection des lignes sur la sphère.

Soit $[P_m; P_n]$ un autre segment définissant avec O^2 le plan Π_{mn}. L'intersection des plans Π_{ij} et Π_{mn} forme une ligne passant par O^2 et l'intersection des segments $[P_i; P_j]$ et $[P_m; P_n]$ dans le cas d'un point de fuite fini (figure 1.7(c)). Dans le cas d'un point de fuite infini, la ligne formée par les deux plans sera parallèle au plan de l'image. Cette intersection de plans est définie par un vecteur calculé grâce à un produit vectoriel entre les deux vecteurs normaux aux plans selon

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \overrightarrow{N_{\Pi_{ij}}} \times \overrightarrow{N_{\Pi_{mn}}}. \tag{1.4}$$

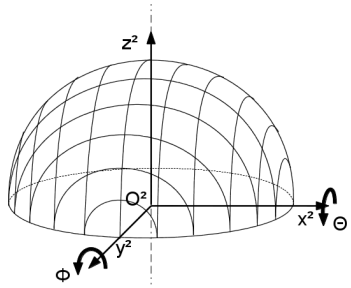


FIGURE 1.8 – Division de la sphère en fonction des angles Φ et Θ

Le vecteur obtenu par la formule précédente permet de définir les angles Φ et Θ. Ils correspondent à l'intersection des deux plans par rapport au repère O^2. Ils sont calculés suivant

$$\Phi = \arctan \frac{z}{x} \quad \text{et} \quad \Theta = \arctan \frac{z}{y}. \tag{1.5}$$

La sphère est alors divisée en cases associées aux angles Φ et Θ (figure 1.8). Chaque couple d'angles Φ et Θ calculés constitue un vote pour la case correspondante. Seules les cases dont le nombre de votes dépasse un certain seuil seront considérées. Les moyennes A_Φ et A_Θ des angles associés à ces cases sont alors calculées : elles sont ensuite utilisées pour retrouver la position du point de fuite sur l'image, suivant

$$x = H \times \tan A_\Phi \ ; \ y = H \times \tan A_\Theta, \tag{1.6}$$

où H est la hauteur du plan image par rapport à O^2.

Afin d'améliorer la robustesse de la technique, un poids est attribué à chaque ligne lors du vote. Ce poids dépend de la taille de la ligne : plus la ligne est grande, plus sa contribution à détecter le point de fuite est importante. Cette solution permet d'éviter de détecter des points de fuite issue de lignes de petites tailles.

1.4.2 Détection sol/mur

Afin de simplifier la recherche des portes dans l'image, nous procédons à une réduction de l'espace de recherche. En effet, les portes sont toujours localisées sur les murs. Une fois que les murs sont détectés, la détection des portes consiste à chercher les formes rectangulaires dans ces seuls espaces "mur".

Dans la littérature, différentes techniques ont été proposées pour résoudre ce problème de séparation sol/mur. Dans [Ok 12], les points d'intersection des lignes verticales et le plan du sol sont utilisés pour la détection de la frontière

sol/mur. Une autre approche statistique dans [Delage 06] prend chaque colonne de l'image à part et classe les pixels partant du bas de l'image vers le haut. La classification consiste à dire si le pixel appartient au sol ou non en utilisant un modèle de réseau bayésien.

Dans notre approche, nous procédons différemment : sachant la structure d'un couloir et la position du point de fuite, l'objectif est de trouver deux lignes de séparation sol/mur de chaque coté du couloir. La figure 1.9 présente le résultat de la détection sol/mur souhaité : le sol (en rouge) se trouve en bas de l'image entre les deux lignes séparatrices. Quant aux murs (en bleu), ils sont délimités par les lignes séparatrices sol/mur.

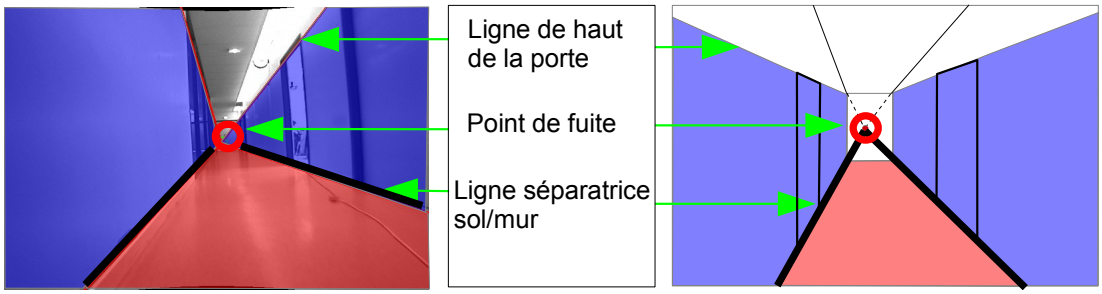


Figure 1.9 – Détection de la structure globale du couloir : les lignes séparatrices sol/mur permettent de détecter les surfaces correspondant au sol (rouge) et murs (bleu)

Dans la suite, nous allons présenter deux techniques pour résoudre ce problème.

Détection basée sur les intersections des lignes verticales

Les deux lignes séparatrices sol/mur sont en intersection avec les extrémités inférieures des lignes verticales (montant de porte, fin de mur). Cette première approche consiste donc en un système de vote : la ligne de séparation sol/mur correspond à la ligne qui s'intersecte avec le maximum de bas de montants portes. Puisque les montants de portes sont identifiés par les lignes verticales, cela revient à dire que la ligne séparatrice sol/mur correspond à une ligne non-verticale en intersection avec un maximum d'extrémités de lignes verticales.

Le nombre d'intersections entre une ligne verticale et les lignes non-verticales est obtenu par le calcul de distance entre la ligne non verticale et le point d'extrémité de cette ligne verticale. Cette distance est limitée par un seuil η fixé empiriquement afin de s'assurer que les lignes sont "vraiment" proches du sol.

La ligne séparatrice est située en dessous du point de fuite : nous ne sélectionnons que les lignes non verticales se trouvant en dessous du point de fuite. De plus, chaque mur du couloir est traité séparément. Pour cela, l'image est décomposée en 4 zones selon la position de point de fuite NO, NE, SO, SE (Nord, Ouest, Sur, Est) comme illustré dans la figure 1.10. Chaque zone inférieure (SO, SE) de l'image devra contenir une ligne de séparation.

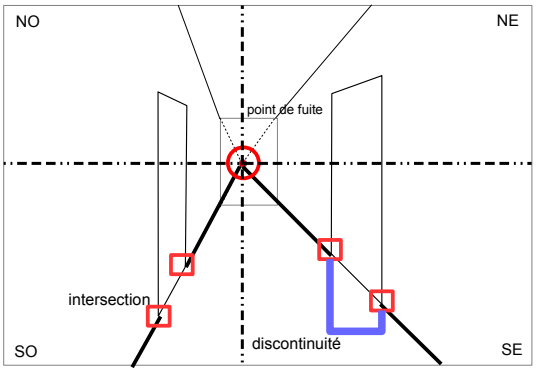


Figure 1.10 – Découpage de l'image en fonction du point et de fuite et calcul des lignes de séparation sol/mur en fonction du nombre d'intersections avec les lignes verticales

La figure 1.11 présente les résultats de la détection de la ligne de séparation sol/mur en utilisant différentes valeurs de η. On peut remarquer que pour une valeur importante de η (η=50), la ligne détectée n'est pas cohérente avec la scène. Par la suite nous fixons $\eta = 10$.

Détection basée sur les lignes de fuite sol/mur

Dans cette seconde solution, nous considérons que les lignes de séparation sol/mur sont des lignes non verticales qui ont forcement contribué au calcul du point de fuite se trouvant au bout du couloir (ce qui est évident puisque elles se trouvent chacune sur un coté du couloir et représentent des lignes parallèles).
Ces deux lignes non verticales doivent être bien apparentes dans l'image et de taille importante par rapport aux autres lignes (placard, haut des portes, radiateur etc). A partir de cette hypothèse, rechercher les lignes de séparation sol/mur, revient à chercher deux lignes avec les propriétés suivantes :

- les deux lignes sont non verticales ;
- les deux lignes ont voté pour le point de fuite détecté ;
- les deux lignes se trouvent au dessous du point de fuite dans l'image ;
- les deux lignes sont de tailles suffisantes (les plus grandes dans l'image).

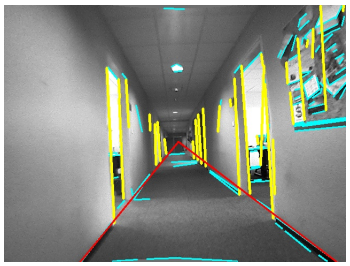
(a) η=5 (b) η=10 (c) η=50

Figure 1.11 – Détection des lignes de séparation sol/mur à travers la technique d'intersection des lignes verticales avec des valeurs de η différentes

Les lignes de séparation sol/mur peuvent être discontinues à cause d'éléments qui peuvent se trouver sur les murs (bas de portes, obstacle). Pour récupérer l'ensemble de la ligne, on procède à une fusion de lignes non verticales se trouvant dans la même partie de l'image (SE,SO) et qui partagent le même angle à un seuil tan_θ près . En procédant ainsi, on peut facilement reconstruire les lignes des bas de mur qui représentent les plus grandes lignes non verticales en direction du point de fuite.

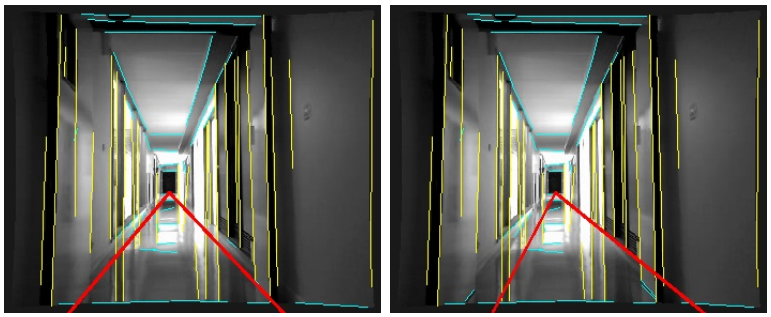
La figure 1.12 présente la détection de la ligne de séparation sol/mur avec $tan_\theta = 0.25$. Les lignes détectées correspondent aux lignes réelles séparant le sol des murs. Dans la figure 1.13 on montre la différence entre les deux techniques lorsque la scène comporte un sol réfléchissant. Dans ce cas, les lignes verticales vont se prolonger sur le sol à cause de leur réflexion. La technique basée sur l'intersection ne pourra pas fournir une bonne détection à cause du seuil fixé : il est difficile de prédire jusqu'à quelle distance les lignes vont se prolonger sur le sol. Par contre, la technique basée sur la détection de la lignes de fuite fournit un bon résultat parce qu'elle est indépendante des lignes verticales : seules les lignes non-verticales sont utilisées et ne sont pas concernées par la réflexion du sol. Ainsi la détection reste cohérente avec les mêmes paramètres que lorsqu'on se trouve dans un sol normal. Cependant cette technique échoue si le sol contient des lignes (carrelage), la détection peut considérer une ligne du sol comme ligne du fuite.

Une fois que la détection des lignes séparatrices a été réalisée, nous allons exploiter ces informations afin d'estimer la position relative de la caméra ainsi que sa rotation dans le couloir. Ces deux informations sont nécessaires à la détection des portes dans le couloir.



(a) tan_θ=0.25

Figure 1.12 – Détection des lignes de séparation sol/mur à travers la technique d'estimation de la ligne de fuite



(a) tan_θ=0.25 (b) η=50

Figure 1.13 – Détection des lignes dans le cas d'un sol réfléchissant : la détection avec la ligne de fuite (a) est plus robuste que celle avec les intersections (b)

1.5 Localisation dans un couloir

Afin de pouvoir se localiser dans le couloir, nous allons exploiter la géométrie de la structure du couloir qui suit les règles standards de construction. En effet, nous supposons que les deux murs du couloir sont parallèles et produisent une perspective en profondeur (cf. figure 1.14.b).

Puisque l'on considère que le plan de l'image est toujours perpendiculaire au sol, cela implique que $\phi_y = 0$ (1.14.a). De plus, le fauteuil roule sur un sol plat, donc $\phi_x = 0$. Donc ce qui reste à estimer est la rotation autour de l'axe z : ϕ_z. La caméra est toujours à hauteur h fixe du sol donc $Z_c = h$. Dans notre contexte, on cherche à estimer la position latérale de la caméra à travers sa coordonnée Y_c. Dans le schéma 1.14.c, sont indiqué le point de fuite vp, les deux lignes sol/mur (en rouge) ainsi que le centre du couloir (en bleu).

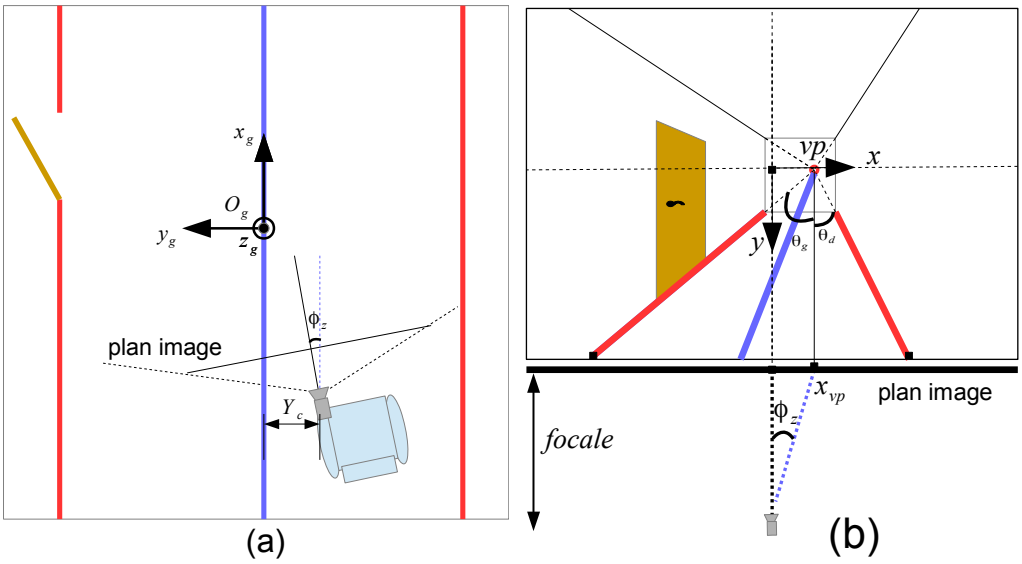


Figure 1.14 – Estimation des paramètres de position et de rotation de la caméra dans le couloir : (a) vue de haut, (b) la perspective du couloir obtenue.

Rotation

Lorsque la rotation de la caméra est nulle ($\phi_z = 0$), on a la coordonnée en x du point de fuite $x_{vp} = 0$. Si le fauteuil effectue une rotation, le point de fuite s'éloigne du point principal de l'image, et $x_{vp} \neq 0$. Ainsi, la rotation ϕ_z se calcule géométriquement selon :

$$\phi_z = atan(\frac{x_{vp}}{p_x}) \tag{1.7}$$

Position latérale

Pour estimer la position Y_c du fauteuil dans le repère O_g, il faut connaître d'abord les angles θ_g et θ_d qui correspondent respectivement aux angles des lignes de séparatrices sol/mur gauche et droite. Ces angles sont extraits directement de la paramétrisation (ρ, θ) des lignes séparatrices sol/mur (cf. figure 1.15).

La translation est définie en fonction des angles θ_g et θ_d. Si la caméra se trouve au centre du couloir ($Y_c = 0$), ces deux angles sont égaux. Ensuite, plus la camera s'approche du mur droit (resp. gauche), plus l'angle θ_d (resp. θ_g) devient petit. En fait, cette position relative par rapport aux tangentes de θ_g et θ_d s'exprime par :

$$Y_c = \frac{tan(\theta_d) + tan(\theta_g)}{tan(\theta_d) - tan(\theta_g)} \frac{Lc}{2} \tag{1.8}$$

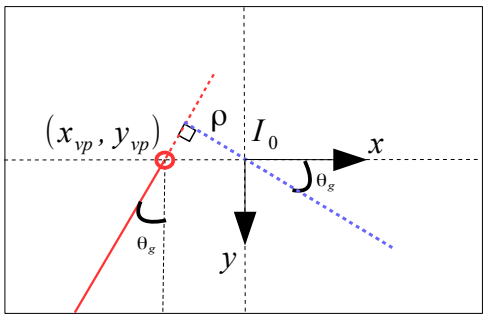


FIGURE 1.15 – Paramétrisation (ρ, θ) d'une ligne séparatrice dans le plan image.

avec Lc la largeur estimée du couloir telle que

$$Lc = h\frac{p_y}{p_x}\left(tan(\theta_d) - tan(\theta_g)\right) \tag{1.9}$$

Preuve : considérons $p_d(x_{p_d}, y_{p_d})$ (resp. $p_g(x_{p_g}, y_{p_g})$) un point appartenant à la ligne séparatrice sol/mur droite (resp. gauche) dans le plan image. Estimer la largeur du couloir revient à estimer la distance entre les coordonnées du monde réel Y_{p_g} et Y_{p_d} des points appartenant au mur.

$$Lc = Y_{p_d} - Y_{p_g}$$

sachant que Y_{p_d} est estimée comme suit (même raisonnement pour Y_{p_g}) :

$$\frac{x_{p_d}}{p_x} = \frac{Y_{p_d}}{X_{p_d}} \Rightarrow X_{p_d} = p_x\frac{Y_{p_d}}{x_{p_d}}$$

$$\frac{y_{p_d}}{p_y} = \frac{h}{X_{p_d}} \Rightarrow X_{p_d} = p_y\frac{h}{y_{p_d}}$$

à partir des deux formules, on peut définir Y_{p_d} comme suite :

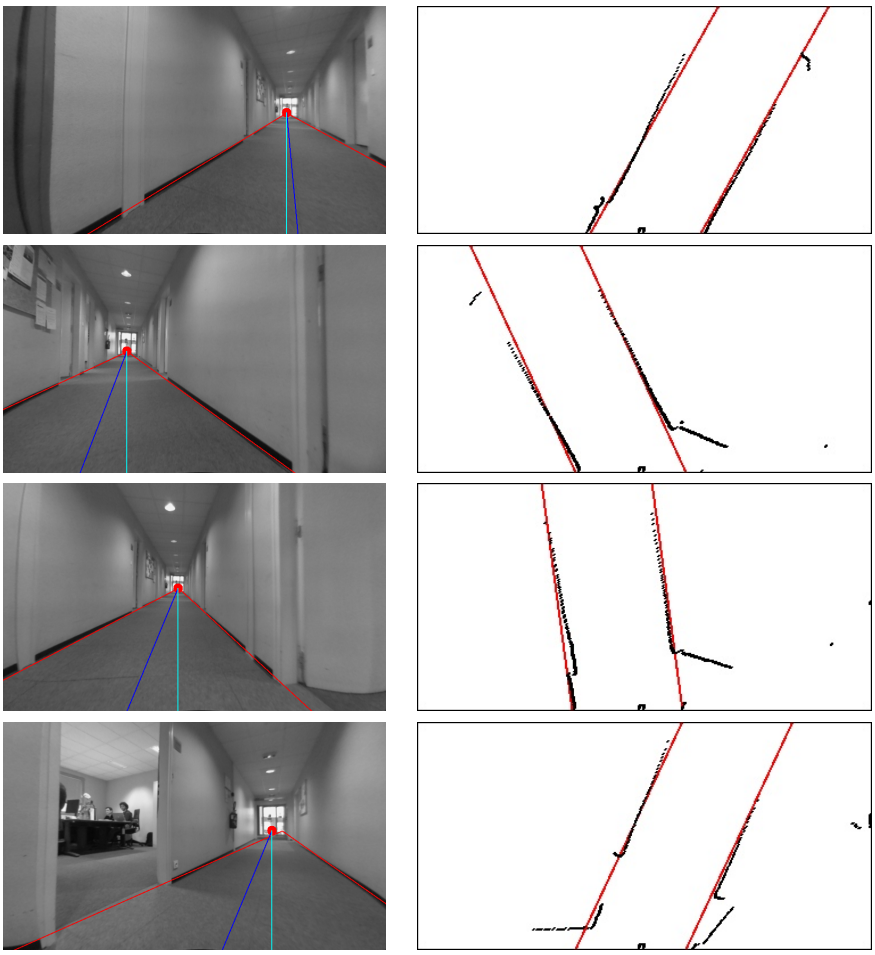
$$Y_{p_d} = h\frac{p_y}{p_x}\frac{x_{p_d}}{y_{p_d}}$$

on remplaçant $tan(\theta_d) = \frac{x_{p_d}}{y_{p_d}}$ on obtient :

$$Y_{p_d} = h\frac{p_y}{p_x}tant(\theta_d)$$

Ainsi, Lc est définit comme suit :

$$Lc = Y_{p_d} - Y_{p_g} = h\frac{p_y}{p_x}(tant(\theta_d) - tant(\theta_g))$$



(a) (b)

Figure 1.16 – Résultats de localisation du fauteuil dans le couloir. (a) les résultats de l'estimation des lignes sol/mur. (b) vue de haut dans le repère caméra : les lignes rouges correspondent à la position et la rotation des murs estimées, les points noirs sont les distances estimées par un laser.

En remplaçant l'équation (1.9) dans (1.8) on obtient :

$$Y_c = \frac{h}{2}\frac{p_y}{p_x}\left(tan(\theta_d) + tan(\theta_g)\right) \tag{1.10}$$

Dans la figure 1.16, les résultats de localisation et d'orientation sont comparés aux résultats obtenus par un laser afin de valider la technique d'estimation proposée dans cette section. Dans la colonne à gauche, on remarque que les points de fuite et les lignes séparatrices sol/mur sont bien estimés ; une bonne estimation de ces deux descripteurs est nécessaire pour avoir la localisation de la caméra. La position et la rotation estimées de la caméra dans le couloir sont transformées dans le repère caméra afin de les comparer avec les données du laser. Ces données sont superposées dans la colonne à droite pour observer l'erreur d'estimation. On peut observer que l'orientation de la caméra est bien estimée en fonction de la position du point de fuite : l'angle de la ligne des points laser correspond à l'angle des lignes estimées. Cependant, on remarque que par exemple dans la troisième ligne, les points lasers sont en léger décalage par rapport à la ligne de fuite : ceci est principalement dû à l'existence des éléments sur le mur. Cela ne pose toutefois pas un problème puisque cela n'affecte pas la rotation de la caméra et son impact sur la position est minime.

En résumé, la position et la rotation estimées de la caméra en fonction des lignes sol/mur ainsi que le point de fuite permettent de nous localiser dans le couloir, et ainsi fournir des informations primordiales pour la reconnaissance des portes. Dans la section suivante, nous allons présenter comment exploiter toutes ces informations afin de détecter les portes dans l'image.

1.6 Reconnaissance des portes

Maintenant que les lignes séparatrices sol/mur sont identifiées et la position et la rotation de la caméra estimées, on peut aborder la détection de portes se trouvant sur les murs. Le processus de détection consiste à retrouver un couple de lignes verticales dans l'image correspondant aux montants de la porte dont la distance dx dans le plan image respecte la largeur de la porte. Donc, pour chaque ligne verticale (premier montant), on estime la coordonnée en x (l'axe horizontal de l'image) de la ligne supposée être le deuxième montant en fonction de :

- la distance de la porte à la caméra,
- la position et l'orientation de la caméra dans le couloir,
- la largeur réelle de la porte.

La figure 1.17 illustre le principe de l'estimation de la position en x du deuxième montant de la porte dans le plan image. On considère les données

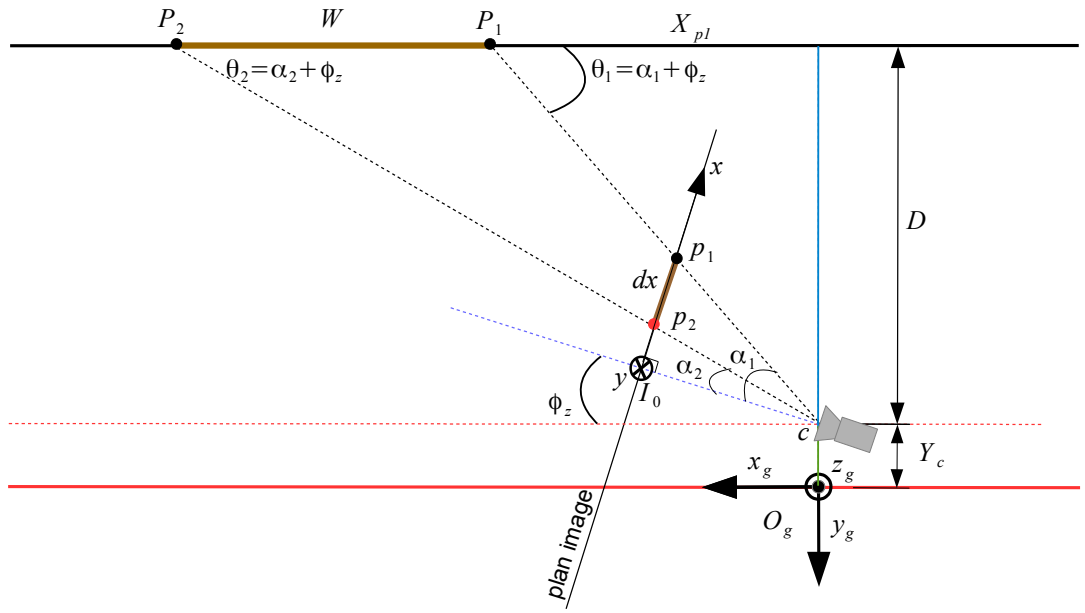


Figure 1.17 – Estimation de la largeur dx de la porte dans le plan image

suivantes :

- D est la distance de la caméra par rapport au mur où $D = |Y_c - Y_{P_1}|$
- P_1 est projeté dans l'image au point $p_1(x_{p_1}, y_{p_1})$
- $P_1(X_{P_1}, Y_{P_1}, Z_{P_1})$ est le point correspondant au premier montant de la porte. Il se trouve au sol et est exprimé dans le repère global. On a donc $P1(X_{P_1}, -\frac{1}{2}Lc, 0)$ avec :

$$X_{p_1} = \frac{D}{tan(\alpha_1 + \phi_z)} \quad \text{avec } \alpha_1 = atan(\frac{x_{p_1}}{p_x}) \tag{1.11}$$

- W correspond la largeur réelle de la porte.
- $P_2(X_{P_2}, Y_{P_2}, Z_{P_2})$ est le point du deuxième montant de la porte. Il se trouve aussi sur le sol impliquant $P_2(X_{P_2}, -\frac{1}{2}Lc, 0)$ avec

$$X_{P_2} = X_{P_1} + W. \tag{1.12}$$

- P_2 est projeté dans l'image au point $p_2(x_{p_2}, y_{p_2})$.

On a :

$$\theta_2 = \alpha_2 + \phi_z = atan(\frac{D}{X_{p_2}}),$$

soit

$$\alpha_2 = atan(\frac{D}{X_{p_2}}) - \phi_z,$$

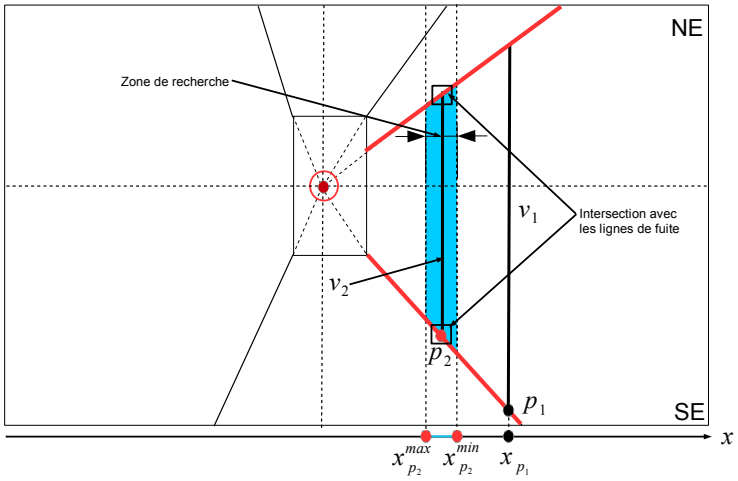


Figure 1.18 – Détection du second montant v_2 dans la zone de recherche en fonction de v_1, $x_{p_2}^{min}$ et $x_{p_2}^{max}$

ou encore

$$tan(\alpha_2) = tan\left(atan(\frac{D}{X_{p_2}}) - \phi_z\right). \tag{1.13}$$

D'autre part, on a :

$$tan(\alpha_2) = \frac{x_{p_2}}{p_x},$$

soit

$$x_{p_2} = p_x tan(\alpha_2). \tag{1.14}$$

La coordonnée x_{p_2} est estimée en remplaçant $tan(\alpha_2)$ par son expression (1.13)

$$x_{p_2} = p_x\ tan\left(atan(\frac{D}{X_{p_2}}) - \phi_z\right) \tag{1.15}$$

$$x_{p_2} = p_x\ tan\left(atan(\frac{D}{X_{p_1} + W}) - \phi_z\right) \tag{1.16}$$

avec X_{P_1} définit par (1.11).

x_{p_2} exprime donc la coordonnée en x du deuxième montant v_2 de la porte supposée connaissant la coordonnée x_{p_1} du premier montant v_1. La figure 1.18 montre la position du point p_2 dans l'image. Il faut maintenant vérifier l'existance d'une ligne verticale dans l'image dont la distance en x est proche du point estimé x_{p_2}. Pour cela, nous définissons une zone de recherche en définissant $x_{p_2}^{min}$ et $x_{p_2}^{max}$ en utilisant les paramètres W^{min} et W^{max}. L'espace de recherche est illustré dans la figure 1.18 en bleu : il délimite l'espace où devra être localisé le

deuxième montant de la porte. Une fois que l'espace de recherche est défini, on commence par chercher des lignes verticales à l'intérieur de cet espace. Il peut s'y retrouver plusieurs lignes verticales de tailles différentes. Afin de s'assurer que la ligne retenue correspond bien à un montant de porte, il faut que cette dernière joigne les deux lignes de fuites (la ligne séparatrice sol/mur et la ligne passant par le haut de la porte) avec ses extrémités. C'est à dire se trouver sur le coté droit du point de fuite, la ligne verticale doit être capable de relier la ligne de fuite qui se trouve dans la partie NE et la deuxième ligne de fuite SE. Si, avec toutes ces contraintes, une ligne verticale est retrouvée, on peut confirmer qu'une forme géométrique respectant les proportions d'une porte a été détectée sur le mur.

Une dernière étape consiste alors à vérifier qu'un contour existe sur la ligne virtuelle qui relie les deux extrémités supérieures des montants de porte. Cette région correspond au haut de la porte et il n'est pas toujours facile de détecter des segments avec l'algorithme LSD. Donc, si on arrive à cette étape à déterminer un gradient non nul autour de cette ligne, on peut affirmer alors que la forme rectangulaire détectée correspond à une porte.

1.7 Résultats

Afin de valider notre approche, un banc de test vidéos a été réalisé. En utilisant le fauteuil électrique et la caméra monoculaire, plusieurs séquences ont été réalisées dans différents couloirs. Les conditions d'illumination changent d'une séquence à une autre. La configuration et le nombre de portes changent aussi : en effet, la couleur de porte varie (jaune, bleu, orange). Parfois les portes et les murs peuvent aussi être de la même couleur (blanc). Ces séquences vont tester la robustesse de notre technique, c'est-à-dire voir à quel point l'algorithme est capable de détecter les portes même dans les pires cas possibles. Les images sont d'une résolution 640x480 pixels. Les expérimentations sont effectuées sur un processeur Intel Core i7 CPU @2.6Hz. Ici nous avons utilisé la technique de détection des lignes séparatrices avec un seuil $tan_\theta = 0.25$. De plus nous avons fixé les paramètres $W^{min} = 0.80m$ et $W^{max} = 1.0m$ selon la taille des portes observée dans les couloirs $W = 0.90m$. En effet, si les valeurs W^{min} et W^{max} sont très proches, on va manquer plusieurs portes. Par contre, si le deux valeurs éloignées, on va augmenter le taux de fausse détection en détectant des formes rectangulaire ne correspondant pas à la porte.

Le tableau (1.1) présente les résultats de l'expérience : la technique s'avère capable de détecter plus de 82% des portes dans le couloir avec un taux de fausse

	images	portes	bonne détection	fausse alarme
total	3414	6356	5212	459

Table 1.1 – Performance de la détection de portes dans différents couloirs

alarme de moins de 13%.

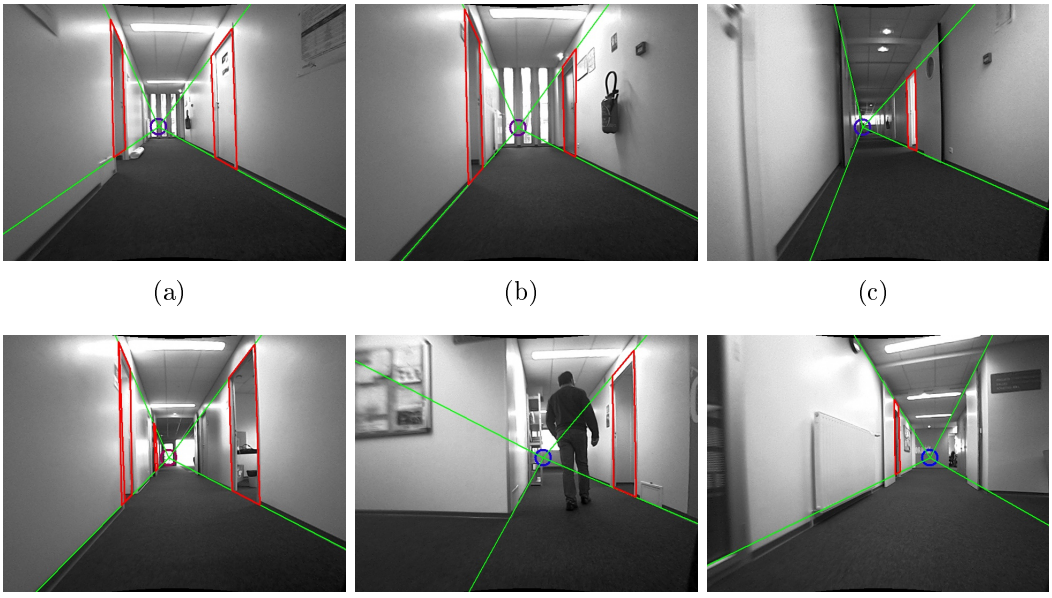
La figure 1.19 montre les résultats obtenus et affiche des portes détectées sous différentes conditions. Dans la plupart des cas, la scène est correctement analysée entraînant une bonne détection des portes. Les figures 1.19(a), 1.19(b), 1.19(c) 1.19(d) montrent de bonnes détections de portes. On peut remarquer que le couloir comporte de multiples portes que l'algorithme est capable de détecter. Dans les figures 1.19(a) et 1.19(b), plusieurs portes sont détectées dans la même image, qu'elles soient ouvertes ou fermées.

La figure 1.19(e) montre la robustesse de l'algorithme en détectant une porte avec un objet en mouvement (une personne) qui n'altére ni l'estimation de point de fuite ni la détection des lignes sol/mur.

Enfin, la détection de porte fonctionne aussi bien dans les croisements de couloir (figure 1.19(e)). La caméra se trouve dans un espace avant le couloir, ce qui n'empêche pas une bonne détection de point de fuite ni des lignes sol/mur.

L'algorithme est donc capable de détecter les portes qu'elles soient ouvertes ou fermées. L'algorithme détecte les portes à des distances variées dans le couloir. Dans notre cas, les portes dans la scène en perspective sont détectées même si leur profondeur est importante : les tests montrent que l'on est capable de détecter des portes à plus de 5 mètres de distance, ce qui est largement suffisant pour entamer une manoeuvre de franchissement de porte.

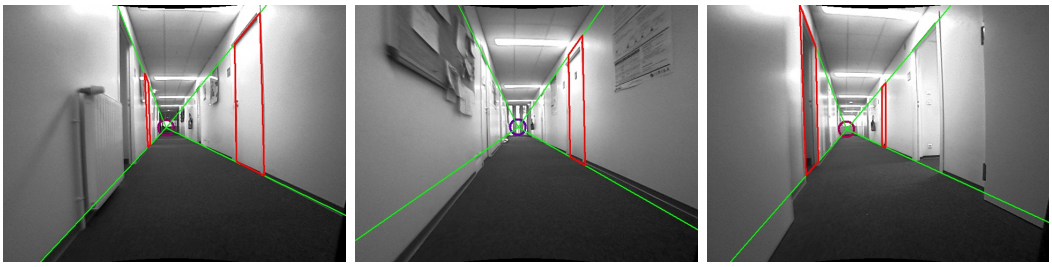
L'algorithme échoue néanmoins dans quelques situations (figure 1.20) : de mauvaises conditions d'illumination entraînent un faible gradient local. Dans cette situation, l'algorithme LSD manque d'informations de gradient et il est donc incapable de détecter les lignes dans l'image. Ceci compromet le calcul du point de fuite, de la ligne séparatrice sol/mur et donc entache la phase de détection de portes. Une phase de pré-traitement peut être utile dans ce cas : augmenter le contraste dans l'image en égalisant l'histogramme de niveau de gris par exemple. Ceci augmente les gradient dans l'image et par conséquent améliore la détection de lignes.



(a) (b) (c)

(d) (e) (f)

Figure 1.19 – Bonne détection de portes



(a) (b) (c)

Figure 1.20 – Échec de détection de portes

1.8 Conclusion

Dans ce chapitre, nous avons présenté un nouvel algorithme de détection de portes dédié à l'initialisation automatique d'un schéma de suivi de portes à des fins de navigation en milieu intérieur.

Les solutions de l'état de l'art imposent souvent des contraintes sur l'état de la scène et celle des portes (caméra face à une porte fermée, porte de couleur différente, etc). Cependant, notre solution est moins contraignante, et utilise des descripteurs géométriques facilement estimables. Elle apporte plusieurs avantages :

– détection de portes ouvertes/fermées ;

- détection de portes de même couleur que les murs ;
- détection de portes à distance de la caméra.

La solution proposée dépend toutefois fortement de l'estimation du point de fuite afin d'extraire le modèle géométrique du couloir et ainsi de la porte. Une mauvaise estimation de ce paramètre due à des conditions non favorables de la scène (mauvais éclairage, obstacle ...) entraîne des échecs de détection dans l'algorithme (détection des lignes séparatrices sol/mur, détection de portes). Néanmoins, ce problème peut être compensé lorsque le fauteuil commence à bouger : les conditions redeviennent alors favorables à une bonne extraction des descripteurs géométriques. Cette solution reste envisageable puisque le mouvement du fauteuil est relativement lent dans un environnement intérieur.

Nous avons aussi présenté une technique de localisation de la caméra dans le couloir. L'idée est d'extraire la position et la rotation de la caméra qui vont servir à estimer la transformation appliquée à la porte dans l'image. Cette technique se base sur un ensemble d'informations issues de la structure régulière d'un couloir. A partir du point de fuite et des lignes sol/mur, on a montré qu'on est capable de déterminer à la fois l'orientation de la caméra dans le couloir et sa position latérale. L'application de cette technique à la navigation du fauteuil a été validée dans [Pasteau 13].

De plus, la solution est de faible coût puisqu'une simple caméra monoculaire suffit contrairement aux solutions utilisant des lasers. Ceci nous offre une solution acceptable pour la mise en œuvre sur le fauteuil.

Maintenant que les portes sont détectées, il va falloir trouver une solution pour pouvoir les suivre dans la séquence d'images, dans le but de garantir une cohérence spatio-temporelle des portes détectées. Pour cela, nous allons proposer dans le chapitre suivant un algorithme de suivi de porte dédié à notre cas d'application.

Chapitre 2

Suivi des portes dans les séquences d'images

Dans ce chapitre, nous allons présenter une technique de suivi de portes. Il s'agit en effet de fournir les informations nécessaires à la tâche de franchissement des portes. Ces informations sont du type position des portes dans le couloir en temps réel. Les challenges sont multiples :

- proposer une solution basée vision,
- trouver les caractéristiques pertinentes à suivre,
- assurer la cohérence spatio-temporelle des portes,
- garantir une faible complexité de calcul.

Le suivi d'objet est une tâche très importante dans le domaine de la vision par ordinateur. Il consiste à estimer le déplacement d'un objet dans la scène par le biais de primitives visuelles qui le représentent. Dans ce chapitre, nous ne nous intéressons qu'aux mouvements rigides (suivi d'objets non déformables).

Le suivi d'objet permet alors d'identifier à chaque instant les objets et caractérise l'évolution de ces objets dans une séquence vidéo à travers l'analyse de leur cohérence spatio-temporelle. De même, il permet de fournir des informations liées à l'objet (orientation, distance) en estimant la transformation qui permet de recaler l'objet sur l'image courante.

La tâche de suivi peut être difficile à cause [Yilmaz 06] :

- du mouvement complexe de l'objet,
- de la forme complexe de l'objet,
- de l'occultation partielle ou totale de l'objet à suivre ;
- d'un changement d'illumination.

La section 2.1 traite ainsi de la problématique de la sélection des primitives visuelles nécessaires au suivi. Dans la section 2.2, nous allons présenter un état de l'art sur les méthodes de suivi d'objets basées vision. Ensuite dans la section 2.3, nous allons présenter notre schéma de suivi de portes. Enfin, la section 2.4 présente les expérimentations effectuées et les résultats obtenus.

2.1 Sélection des primitives visuelles

Le choix des primitives visuelles à suivre est primordial pour assurer un bon suivi. En effet, ce choix est fortement lié au domaine d'application et de la forme de l'objet. La littérature est très riche à ce propos [Teulière 10] : seuls les éléments essentiels sont présentés dans cette section.

2.1.1 Couleur

La couleur est un premier descripteur visuel simple capable de distinguer plusieurs objets dans une scène. L'espace couleur RVB (Rouge, Vert Bleu) est un espace classiquement utilisé pour représenter la couleur d'un pixel dans une image. Le problème avec cet espace couleur vient du fait que les trois composantes sont corrélées. Il existe un autre espace de couleurs YUV très largement répandu qui permet de décorréler la luminance Y des deux composantes de chrominance UV [CCITT 92]. Dans la suite, nous allons utiliser le YUV pour effectuer le suivi des portes.
Cette description au niveau pixel permet de détecter les variations qui peuvent intervenir dans l'image. Pour cela, on peut utiliser des fonctions de similarités telles que la minimisation de la SSD (Somme des différences au carré) [Hager 98a, Benhimane 07] ou en utilisant l'information mutuelle dérivée de l'entropie de l'image [Dowson 08, Dame 10].

2.1.2 Contours

Les contours des objets forment des structures induisant un fort changement d'intensité lorsqu'un mouvement se produit. Une des propriétés principales des contours est qu'ils sont moins sensibles aux changements d'illumination au contraire des primitives basées sur la couleur. Il existe plusieurs détecteurs de contours ; le plus utilisé est le filtre de Canny [Canny 86] capable d'extraire les contours d'une image. Différentes approches ont été proposées dans la littérature : par exemple [Yokoyama 05] combine l'information de contour avec l'estimation de mouvement pour suivre les objets détectés. Une autre approche bayesienne combinant à la fois les informations texture et couleur pour suivre les contours a été proposée dans [Yilmaz 08].

2.1.3 Primitives visuelles basées sur le mouvement : Flot optique

Le flot optique consiste à estimer le vecteur de déplacement pour chaque pixel entre deux images successives. L'hypothèse est que la couleur d'un pixel d'une image est indépendante du temps. Cela revient à dire que la dérivée de la fonction couleur $I_t(u, v)$ au point de coordonnées (u, v) le long du flot optique est nulle, selon

$$\forall x = (u, v), \frac{dI_t(x)}{dt} = \frac{\partial I_t(x)}{\partial t} + \frac{\partial I_t(x)}{\partial u}\frac{\partial u}{\partial t} + \frac{\partial I_t(x)}{\partial v}\frac{\partial v}{\partial t} = 0 \tag{2.1}$$

Cette équation ne permet pas de déterminer de manière unique le flot optique et provoque donc une ambiguïté du mouvement apparent ce qui signifie l'impossibilité d'estimer le mouvement local d'un point sans prendre en compte son voisinage. Pour résoudre cela, [Horn 81] fut l'un des premiers à proposer une formulation qui consiste à ajouter des contraintes de continuité spatiale sur les niveaux de gris ou les champs de vitesse. [Mémin 02, Hilsmann 07] proposent des approches hiérarchiques d'estimation de mouvement. Certaines approches tentent d'estimer le flot optique pour les régions occultées autour des contours [Ince 08, Zhang 12]. Ces techniques sont très intéressantes pour estimer la trajectoire 2D des pixels d'un objet. Cependant, estimer le flot optique dense peut être coûteux en temps de calcul rendant le suivi temps réel impossible.

2.2 Techniques de suivi d'objets

Faire le suivi d'un objet revient à rechercher l'objet d'une image à une autre et donc estimer la trajectoire de l'objet en fonction des propriétés intrinsèques de l'objet à suivre (forme, couleur, texture). Les techniques peuvent imposer des contraintes sur le mouvement et/ou la forme de l'objet. Par exemple, on peut supposer que le mouvement est lisse et sans changement abrupt. Des a priori du modèle de mouvement peuvent donc être injectés dans le suivi (de type vitesse/accélération constante) permettant de prédire la localisation de l'objet dans l'image suivante.

Le suivi d'objet a été largement abordé dans la littérature : plusieurs approches ont été proposées variant les primitives à suivre, et les représentations d'objets utilisées. Les techniques de suivi d'objets peuvent être divisées en deux classes principales : suivi 2D et suivi 3D.

2.2.1 Suivi 2D

Dans le cadre du suivi 2D, l'algorithme cherche à estimer le déplacement dans une image d'un ensemble de primitives visuelles de type descripteurs géométriques 2D (points [Shi 94a, Lucas 81a], segments [Boukir 98, Hager 98a], ellipses [Vincze 01]) ou bien de type contours [Berger 94, Panin 06].

2.2.1.1 Suivi de points

L'objet est représenté par un point (par exemple son centroïde) [Rangarajan 91]. Il peut aussi être représenté par un ensemble de points d'intérêts [Lucas 81a, Harris 88, Shi 94a] qui se caractérisent par une forte variation bi-directionnelle du signal dans le voisinage du point. Il en résulte un suivi plus robuste.

Par exemple, l'algorithme de Kanade-Lucas-Tomasi (KLT) [Lucas 81b, Tomasi 91, Shi 94b] consiste à recaler un modèle d'un objet représenté par des points dans une image. Il est basé sur une méthode de gradient et consiste à chercher la région qui minimise l'erreur avec le modèle recherché. Une description détaillée de l'algorithme KLT est présentée dans la section 4.2.3. Les points utilisés dans le KLT doivent être caractérisés par une forte variation en translation horizontale et verticale dans le plan image. Aussi, l'image doit être texturée afin de bien pouvoir localiser les point suivis. Le KLT présente l'avantage de réaliser un suivi de point très rapide. Cependant, si on utilise un ensemble important de points dans la même image pour calculer le mouvement, le suivi devient alors naturellement plus lent.

Une des propriétés intéressantes du KLT est qu'il permet de suivre des objets rigides (voitures [Cao 11, Pletzer 11]) ou déformables (visage [Ngo 08, Bins 09], personnes [AlNajdawi 12]).

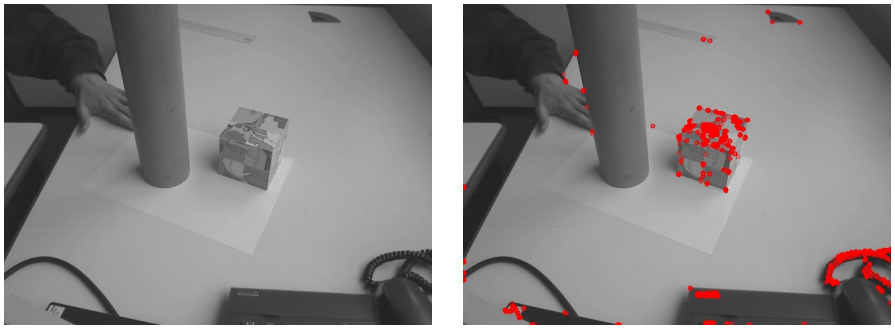


Figure 2.1 – Exemple d'extraction des points d'intérêt de type Harris

2.2.1.2 Suivi de primitives géométriques

Le suivi des primitives géométriques consiste à suivre l'ensemble des points de contours qui les composent. Ensuite, les paramètres de la primitive géométrique suivie sont réestimés offrant alors les nouvelles informations de position et d'orientation.

Pour chaque point de contour, le suivi consiste à rechercher dans l'image suivante le point qui correspond au mieux au point de contour courant. Cette recherche est typiquement effectuée le long de la normale au contour [Marchand 05, Pressigout 06]. Un point de contour est défini par une discontinuité dans le signal : l'objectif est alors de trouver les points à forts gradients se trouvant sur la normale en fonction d'un certain seuil. On peut aussi ajouter quelques contraintes supplémentaires connaissant les propriétés de l'objet à suivre (orientation du gradient notamment). Dans [Teulière 10], un suivi multi-hypothèse est appliqué pour le suivi d'objet. En effet, plusieurs hypothèses correspondant à des contours vraisemblables dans l'image sont générées. Ensuite, les hypothèses sont classées en utilisant un algorithme de k-moyennes pour identifier quel contour correspond le mieux à celui recherché.

Le choix du nouveau point de contour peut alors s'appuyer sur plusieurs stratégies différentes. On peut ainsi sélectionner le premier point de gradient qui satisfait un certain seuil. Aussi, on peut choisir le point avec le gradient le plus élevé ou bien le point similaire en termes de couleur au point de contour de départ.

L'estimation du gradient le long de la normale s'effectue à travers la convolution de l'image avec masque M (cf. figure 2.2). Le masque M doit être orienté en fonction de la normale au contour. Dans une image I_t, on estime la réponse de la convolution avec le masque M au point p_t. Ensuite, la technique consiste à rechercher un point q_j dans I_{t+1} situé sur la normale passant par le point p_t qui maximise la réponse du signal après la convolution avec M.

Après avoir identifié les points de contours, la primitive géométrique est reconstituée de manière à respecter les propriétés de cette dernière. Par exemple, si le contour suivi est une ligne, chaque point (x_i, y_i) obtenu appartenant à la ligne doit vérifier l'équation suivante :

$$x_i a + y_i b + c = 0, \tag{2.2}$$

avec $(a, b, c) \in \mathbb{R}^3$.

Soit X la matrice $n \times 3$ des coordonnées des points de la ligne obtenue avec n le nombre des points de contours. On note

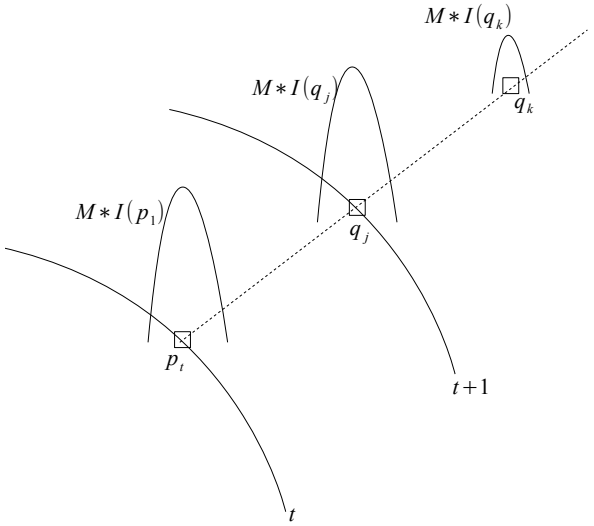


Figure 2.2 – Recherche du point de contour le long de la normale par convolution avec le masque M.

$$\mathrm{X}= \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}.$$

Pour estimer les coefficients $C = (a, b, c)$ de la ligne, la résolution du système

$$XC = 0_{n\times 1} \tag{2.3}$$

peut se faire à travers une technique RANSAC [Fischler 81] qui permet d'éliminer les intrus (outliers) qui sont dûs à des mauvaises estimations.

Le suivi des primitives géométriques basées contours est donc une méthode rapide de suivi et offre l'équation des points contours suivis. Cette technique est aussi robuste vu qu'elle applique un suivi sur des points de contours à fort gradient et reste donc moins sensible aux changements d'illuminations.

Cependant, la vitesse de calcul dépend fortement du nombre de points de contours utilisés pour le suivi de points. En effet, plus le nombre de points d'une ligne augmente, plus le nombre de convolutions avec le masque M augmente. De plus, la complexité dépend aussi de la taille de la fenêtre de recherche des candidats le long de la normale.

2.2.2 Suivi 3D basé modèle

Dans cette catégorie, le suivi d'objet consiste à estimer la position et le déplacement de la caméra par rapport à l'objet. Dans ce cas là, l'objet est représenté par son modèle 3D à travers un ensemble de segments qui définissent la structure volumétrique de l'objet à suivre [Coutard 11, Petit 12]. L'estimation de la position de la caméra consiste à estimer la pose de cette dernière en se basant sur les descripteurs géométriques de l'image (points, lignes, ...).

L'initialisation d'un algorithme de suivi 3D constitue la phase la plus critique dans le processus. En effet, projeter un objet 3D dans une image 2D cause une perte d'informations et crée des faces visibles et des faces cachées. L'initialisation consiste donc à détecter les primitives visuelles (points, lignes, ...) appartenant aux faces visibles de l'objet et les associer à leur correspondant dans le modèle 3D.

Dans [Petit 12], l'initialisation est interactive par la sélection des points dans l'image. Ensuite, le calcul de pose se fait à travers la technique de correspondance de points 2D/3D [Dementhon 95]. Dans [Coutard 11], l'initialisation se fait par une technique de mise en correspondance de patchs pour trouver la pose initiale.

2.2.3 Discussion

Il existe bien d'autres techniques qui n'ont pas été abordées ici puisqu'elles ne sont pas en lien avec ces travaux de thèse. Pour plus d'information, le lecteur peut se référer à [Yilmaz 06].

Dans cette section, nous avons donc présenté un ensemble de techniques de suivi d'objets qui se différencient par leur représentation des objets (2D, 3D) et par les primitives visuelles utilisées pour estimer la trajectoire de l'objet (points, lignes, contours ...). Dans notre contexte de suivi de portes, nous allons représenter les portes par leur montants soient deux segments verticaux identifiés lors de la phase détection des portes. Donc, pour le suivi, nous allons appliquer un suivi de lignes adapté à notre problème en utilisant les a priori dont on dispose.

2.3 Application au suivi de portes

Lors de la détection, nous avons vu à quel point les montants des portes contribuent à la détection de porte. Ceci est dû à leur détection robuste lors de la phase d'extraction des lignes grâce à leur fort gradient, ce qui a permis

leur détection par l'algorithme LSD. Nous allons suivre la même logique pour le suivi de porte. En effet, les montants de porte vont préserver leur propriétés de fort gradient le long de la séquence. Suivre les portes revient donc à suivre les segments correspondant à leur montants. Pour cela nous allons utiliser une technique adaptée au suivi de lignes.

Combiner la détection de portes avec leur suivi n'est pas une tâche facile. En effet, il faut prendre en compte le fait que le système ne dispose pas de carte du couloir et donc il ignore les positions et le nombre de portes dans le couloir. Il faudra donc lancer la détection des nouvelles portes tout en suivant celles qui sont détectées à mesure que le fauteuil avance dans le couloir. Pour cela, nous allons tout d'abord enrichir l'algorithme de détection de portes avec l'ajout de la cohérence spatio-temporelle des descripteurs visuels (points de fuites, lignes sol/mur) dans le but de stabiliser les descripteurs tout au long de la séquence et ainsi minimiser au mieux les erreurs d'estimation des descripteurs.

2.3.1 Cohérence spatio-temporelle des descripteurs visuels

Lorsque le fauteuil roulant roule dans un couloir, les descripteurs vont former une trajectoire lisse et cohérente avec le mouvement de la caméra jusqu'à ce qu'ils sortent du champ de vision. Les outils présentés dans la section 1.3 permettent d'extraire des descripteurs de type lignes ou points de fuite sous les conditions fournies par une image à l'instant t. Or le bon déroulement de cette estimation peut être altéré par plusieurs facteurs :

- faible éclairage,
- objet en mouvement,
- occultation du descripteur.

Dans ces cas là, l'extraction des descripteurs en intra[1] peut donner des résultats faussés entraînant une mauvaise de détection de porte ou l'impossibilité de détecter une porte. Pour pallier ce problème, nous appliquons un filtre temporel afin de corriger les paramètres des primitives géométriques.

Notations

- I_t : image à l'instant t,
- vp_t : point de fuite à l'instant t estimé en intra,
- $\widehat{vp_t}$: point de fuite à l'instant t après filtrage temporel,

1. Extraction en intra : extraction à l'intérieur de la même image

- P_t : point appartenant à une ligne à suivre à l'instant t,
- $Q_{t+1} = \{Q^j_{t+1}\}$: ensemble des points appartenant à la normale de la ligne à suivre à l'instant $t+1$,
- G^j_{t+1} : gradient au point Q^j_{t+1}.

Principes

Les points de fuite vp_t sont estimés en intra à chaque image t de la séquence. Afin d'assurer la cohérence spatio-temporelle entre les points de fuite, on estime $\widehat{vp}_t$ à travers un filtre passe bas le long de la séquence selon :

$$\begin{cases} \widehat{vp}_0 & = vp_0 \\ \widehat{vp}_t & = \alpha \times \widehat{vp}_{t-1} + (1-\alpha) \times vp_t \end{cases}$$

avec $0 \leq \alpha \leq 1$.

Le paramètre α est fixé empiriquement pour assurer un lissage de la variation des points de fuite estimés $\widehat{vp}_t$ dans l'image courante I_t en fonction de la position du point de fuite précédent $\widehat{vp}_{t-1}$ dans l'image I_{t-1} et celui calculé en intra vp_t.

2.3.2 Suivi 2D des portes

Afin de proposer une solution rapide et efficace à notre problème, il faut analyser de près notre cas d'application. En effet, la caméra est fixée sur le châssis du fauteuil roulant : les déplacements de la caméra suivent donc les déplacements du fauteuil. On a montré dans le chapitre précèdent que l'axe optique de la caméra est fixe et horizontal (parallèle au sol). Ceci reste vrai lorsque le fauteuil est en déplacement. Donc les lignes verticales qui correspondent aux montants des portes vont préserver cette propriété de verticalité. Cela nous amène à la conclusion suivante : les lignes verticales vont avoir un déplacement horizontal dans l'image avec des vitesses variées liées au mouvement du fauteuil.

Puisque les portes sont de formes rectangulaires, les montants doivent être toujours parallèles tout au long de suivi. Ces contraintes et hypothèses vont nous permettre de définir une technique robuste de suivi de lignes à complexité réduite.

On sait que les montants des portes vont se déplacer horizontalement, l'objectif est donc de définir la normale aux segments verticaux (horizontale) et chercher la ligne qui répond au critère de gradient. Notre algorithme est inspiré par la technique des contours mouvants (ME : Moving Edges) présentée dans [Bouthemy 89].

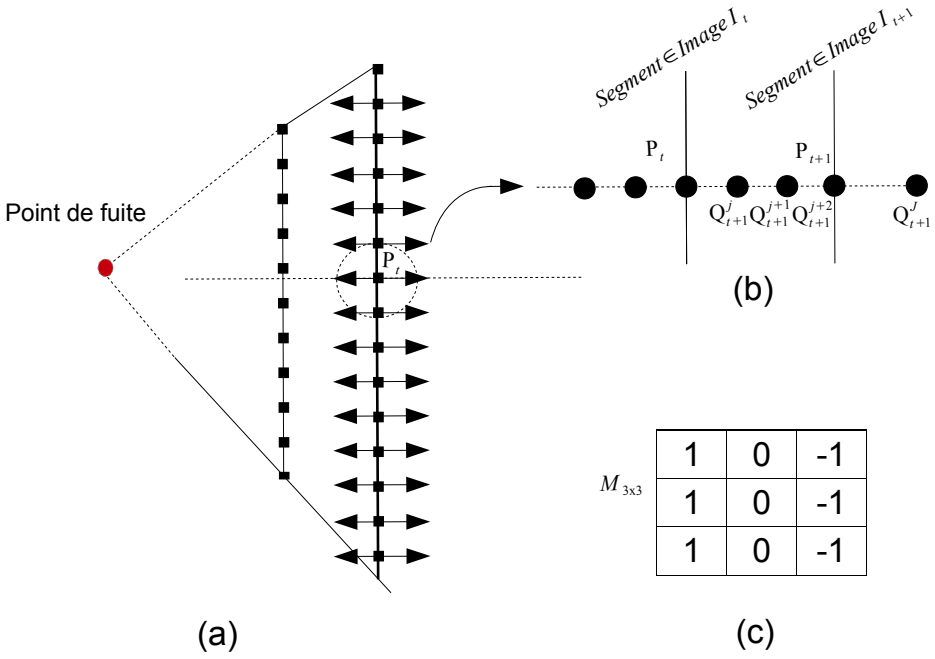


Figure 2.3 – Suivie de portes basé sur ME : (a) définir la normal sur la ligne de la porte. (b) déterminer la position des points de contour dans l'image suivante. (c) la matrice de convolution

Considérons P_t un point dans l'image I_t appartenant à une ligne à suivre. Le processus consiste à chercher son correspondant P_{t+1} dans l'image I_{t+1} qui doit se trouver sur la normale de la ligne. Pour ce faire, une recherche bidirectionelle sur la normale horizontale est effectuée. La figure 2.3 présente le schéma de suivi utilisé dans notre algorithme. On définit pour cela un paramètre J qui précise la distance sur laquelle on va aller chercher le point P_{t+1} sur la normale. L'ensemble $Q_{t+1} = \{Q^j_{t+1}, j \in [-J, J]\}$ correspond aux points se trouvant sur la normale des deux cotés du point P_t.

Ensuite, pour chaque point Q^j_{t+1}, on calcule le gradient local G^j_{t+1} en appliquant une convolution (opérateur $*$) avec le masque orienté M dans le but d'estimer le gradient local. Le masque M est un masque vertical $n \times m$ (cf. figure 2.3.c). En pratique, on a $n < m$ parce que les lignes suivies sont verticales. Le gradient est estimé pour tous les points se trouvant sur la normale. Ainsi, P_{t+1} correspond au point Q^{j*}_{t+1} qui maximise le gradient dans la limite du paramètre J. Le choix du meilleur point Q^{j*}_{t+1} se fait suivant :

$$P_{t+1} = Q_{t+1}^{j*} = argmax_{j\in[-J,J]} G_{t+1}^{j} \tag{2.4}$$

avec

$$G_{t+1}^{j} = |I_{t+1}^{v(P_t)} * M + I_{t}^{v(Q_{t+1}^{j})} * M| \tag{2.5}$$

où $v(.)$ est le voisinage autour du pixel considéré.

Estimation du déplacement de la ligne

Une fois que les déplacements ont été estimés pour chaque point, il faut recalculer les paramètres de la ligne dans l'image I_{t+1}. Sachant que la ligne reste toujours verticale, il est donc inutile de changer son orientation. Nous appliquons ainsi un système de vote (cf. figure 2.4) capable de fixer le déplacement d de la ligne en fonction du nombre de déplacements obtenus pour chaque point de contours. On calcule

$$d = d_{max} \tag{2.6}$$

avec d_{max} la distance qui apparaît le plus dans les déplacements du point $P_t \rightarrow P_{t+1}$. Une fois que le déplacement d est défini, nous changeons les paramètres de position de la ligne dans l'image. Nous appliquons un déplacement par montant de porte à cause de l'effet de profondeur : le montant le plus proche de la caméra aura donc un déplacement *légèrement* plus important que celui qui est plus loin.

Mise à jour des extrémités

Plus les portes s'approchent (s'éloignent), plus leurs formes deviennent de plus en plus grandes (petites) par effet de zoom. Donc les lignes correspondant à leurs montants vont changer de taille à mesure qu'elles s'approchent ou s'éloignent de la caméra. Pour cela, il faut mettre à jour leurs extrémités en fonction du déplacement.

A cet effet, nous appliquons le même calcul de gradient le long de la ligne. Ainsi, nous allons rechercher un point qui appartient à la ligne et qui a le même gradient. La figure 2.5 montre le processus de mise à jour des extrémités de la ligne. Partant des points d'extrémités ext_t^1 et ext_t^2, on applique une convolution avec le masque M sur tous les points de la ligne se trouvant dans l'intervalle R à proximité d'une extrémité de la ligne. Si la ligne devait se prolonger, alors la réponse de la convolution va rester non nulle sur les points de la ligne se trouvant sur la $direction1$ jusqu'à atteindre un point avec une valeur nulle de gradient. D'un autre coté, si la ligne devait se rétrécir, alors, la réponse de la convolution sera nulle pour les points de la ligne se trouvant sur la $direction2$ jusqu'à atteindre un point avec une valeur non nulle du gradient. On change la

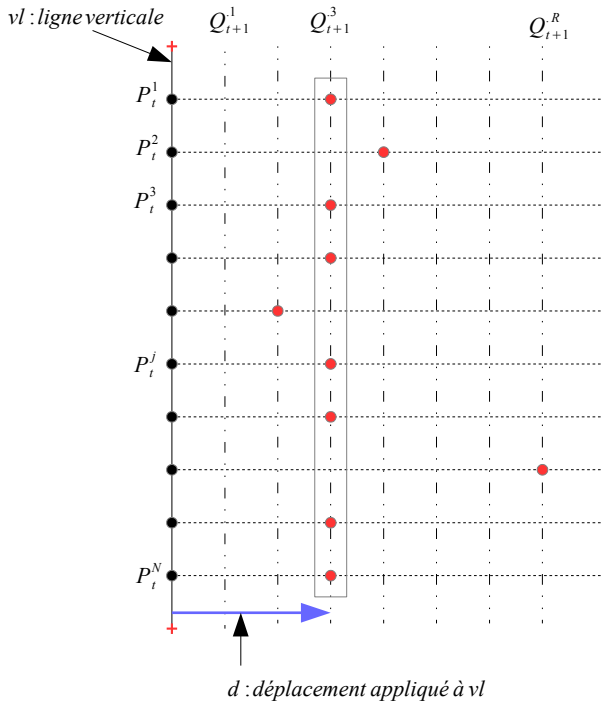


Figure 2.4 – Estimation du déplacement en appliquant le système de vote sur l'ensemble des points de contours de la ligne.

taille de la ligne en changeant les extrémités avec les nouveaux points trouvés ext_{t+1}^1 et ext_{t+1}^2.

Lors de la détection de plusieurs portes, nous lançons un suivi pour chaque porte séparément. On associe alors un suivi de ligne pour chaque montant de porte. A mesure que le fauteuil avance dans le couloir, la porte suivie devient de plus en plus proche jusqu'à ce que le montant le plus proche de la caméra sort du champ de vision. Afin de poursuivre le suivi, notre schéma désactive le suivi de la ligne cachée et continue à suivre le montant visible jusqu'à ce qu'il sort à son tour du champ de vision.

2.4 Expérimentation et résultats

Afin de valider notre approche, nous avons intégré notre suivi de porte dans notre schéma de détection afin d'obtenir une solution complète : détection et suivi de portes. La figure 2.6 montre le schéma général obtenu après fusion des deux techniques. Tout d'abord, lors du lancement de l'application, on passe par une première étape d'initialisation. L'initialisation consiste à calculer les

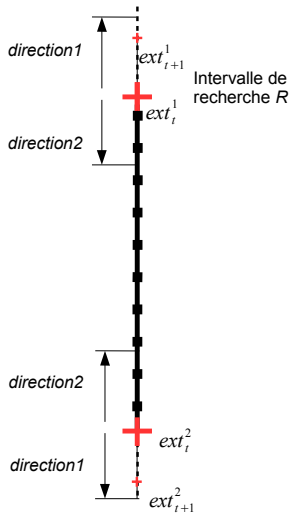


FIGURE 2.5 – Mise à jour des extrémités des lignes.

descripteurs géométriques dans la scène et faire une première détection de portes. Afin de stabiliser le point de fuite et les lignes séparatrices sol/mur, on applique un filtrage temporel sur ces descripteurs afin d'améliorer leurs paramètres. Ensuite, si des portes sont détectées, alors on initialise un suivi pour chaque porte ; sinon on relance l'algorithme de détection jusqu'à trouver la première porte tout en appliquant le filtrage temporel.

Puisqu'on ne connaît pas le nombre de portes et leur localisation dans le couloir, il faudra relancer l'opération de détection toutes les N images. Cette opération permet de détecter l'ensemble des portes (nouvelles portes, ou celles déjà connues). Pour les nouvelles portes, on va leur attribuer un nouveau suivi de porte. Quant aux portes déjà détectées (et en cours de suivi) nous allons utiliser leurs informations (position, taille des lignes) pour vérifier si le suivi est en cohérence avec la forme de la porte détectée.

Les résultats obtenus sont affichés dans la figure 2.7. Nous montrons dans cette figure l'évolution du suivi des portes à travers le suivi de leur montants.

La première ligne correspond à la séquence N°1. Cette séquence est caractérisée par une faible illumination dans le couloir et une forte illumination émanant des bureaux. Ceci offre un très bon contraste donc un bon gradient qu'on peut suivre assez facilement. On remarque dans la figure 2.7(a) que seule la porte à droite a été détectée au départ et le suivi associé a été initialisé et lancé. Ensuite, on remarque qu'après quelques images, la deuxième porte (en face) vient d'être détectée à son tour et son suivi a été lancé. Le fauteuil avance dans le couloir

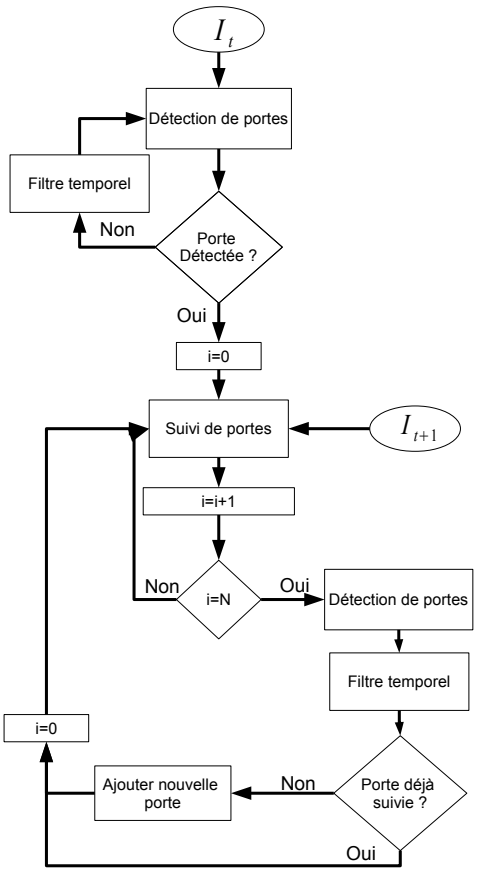


Figure 2.6 – Schèma général de détection et suivi de portes

impliquant un déplacement des lignes des montants de porte dans l'image. Les lignes sont délimitées par des croix rouges correspondant à leurs extrémités. A la dernière image de la séquence 2.7(d), on remarque que seul un montant sur deux est visible et suivi pour les deux portes. Ceci est dû au fait que les portes sont très proches de la caméra et donc en dehors de son champ de vision.

Dans la deuxième séquence, les portes sont de la même couleur que les murs. Ceci complique un peu la tâche de suivi vu que le gradient est très faible. On remarque ainsi que les extrémités supérieures sont recalées avec le plafond et non pas le haut de la porte. Ceci n'est pas vraiment problématique vu que pour la tâche de navigation on va essayer d'éviter la collision avec les montants et non pas le haut de porte.

La troisième séquence nous montre un problème de suivi rencontré lors de nos expérimentations. En effet, pour certaines configurations, un petit mur se trouve juste derrière la porte correspondant à un placard à l'intérieur du bureau

(cf. figure 2.7(k)). Lorsque la porte est ouverte, il peut arriver que la ligne suivie se recale sur le petit mur du placard induisant une erreur. Pour résoudre ce problème, une solution consiste à vérifier lors de la prochaine détection des portes si les lignes s'apparentent avec la porte détectée et les corriger.

Des vidéos sont disponibles sur le site web de l'équipe de Lagadic [2]. Elles permettent de mieux visualiser l'évolution du suivi dans le couloir.

Les résultats obtenus lors de nos expérimentations sont donc très satisfaisants et encourageants. En effet, le suivi de ligne a montré sa robustesse malgré sa simplicité. Le suivi de porte est caractérisé par une faible complexité. Les portes détectées sont suivies en temps réel. La complexité de calcul dépend fortement du nombre de points de contours utilisés pour suivre les lignes. Des expérimentations ont montrées que même si on utilise la ligne complète sans sous-échantillonnage, on reste toujours dans un contexte temps réel. Ceci répond à nos contraintes de départ et donc on pourra envisager à terme cette solution dans le module embarqué du fauteuil roulant.

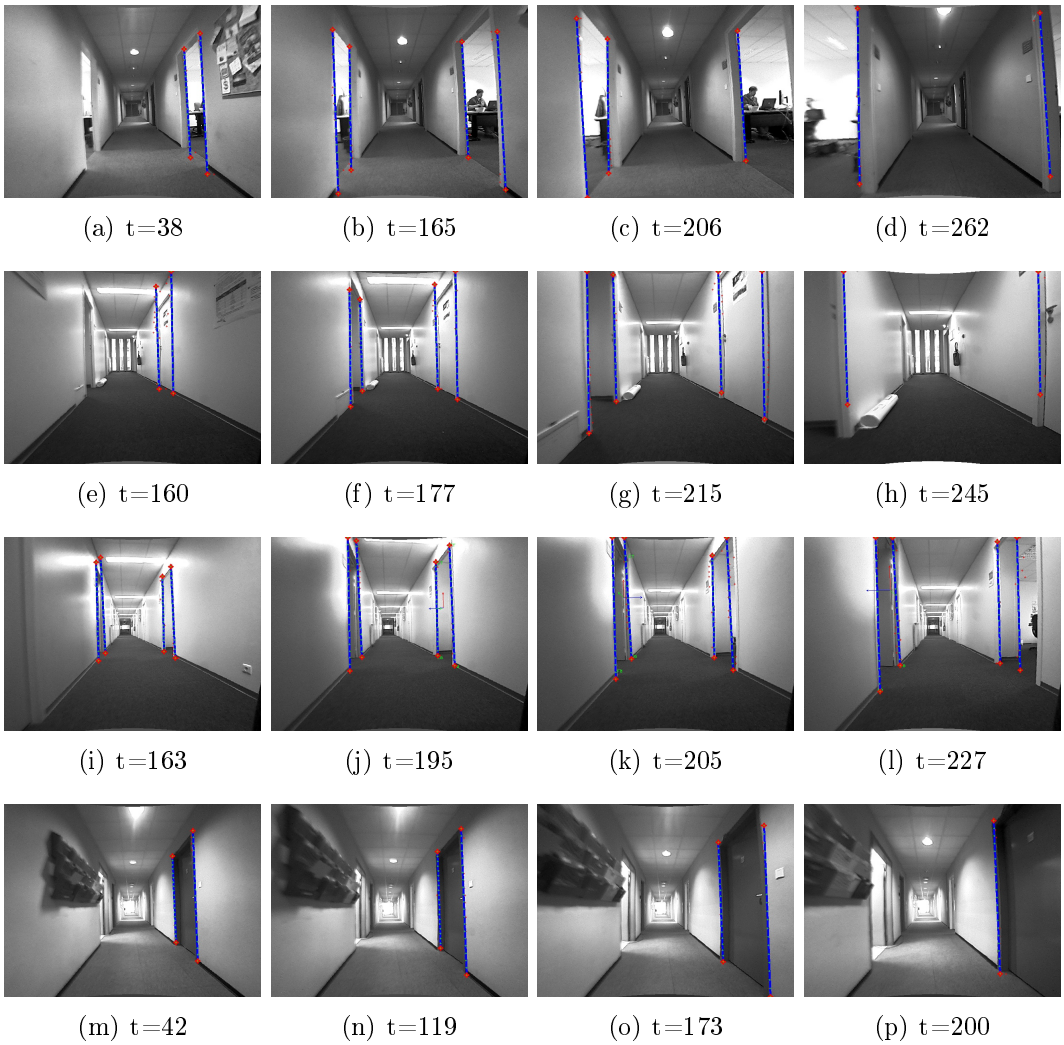
2.5 Conclusion

Dans ce chapitre, nous avons présenté une technique de suivi de portes basée gradient. Pour ce faire, nous avons représenté la porte par deux lignes droites correspondant à ses montants. On a appliqué ensuite une technique de suivi de primitives visuelles basée contours. Cette technique permet d'estimer les paramètres de la ligne en décomposant le problème. En effet, la ligne est représentée par plusieurs points de contours : un vecteur de déplacement est alors estimé pour chaque point de contour le long de la normale à la ligne. Ensuite, nous réévaluons le déplacement global en appliquant un système de votes.

Les résultats obtenus lors de nos expérimentations ont montré l'efficacité de notre technique de suivi. On a montré aussi que les choix opérés sont tout à fait justifiés : suivre la porte en fonction de ses montants et non pas par sa couleur s'est révélé un choix judicieux compte tenu des propriétés de la scène observée.

Notre objectif de départ était de proposer une solution de détection et de suivi de porte dans un environnement intérieur inconnu afin d'aider le fauteuil à franchir les portes. La combinaison des techniques de détection et de suivi nous offre une solution complète capable de s'initialiser automatiquement. Ces

2. http ://www.irisa.fr/lagadic/team/rafiq.sekkal-eng.html



(a) t=38 (b) t=165 (c) t=206 (d) t=262

(e) t=160 (f) t=177 (g) t=215 (h) t=245

(i) t=163 (j) t=195 (k) t=205 (l) t=227

(m) t=42 (n) t=119 (o) t=173 (p) t=200

Figure 2.7 – Résultat du suivi de portes dans différents couloirs sous des conditions d'illumination différentes

travaux ont fait l'objet d'un article [Sekkal 13].

Maintenant que nous avons réussi à détecter et suivre des objets bien connus (portes), nous allons essayer de proposer une solution de détection et de suivi d'objets inconnus dont on ignore la forme, la couleur et le modèle de mouvement. Il s'agit donc de traiter des objets complexes déformables composés de plusieurs modèles de trajectoire ce qui rend leur détection et leur suivi difficiles. Dans le chapitre suivant, nous allons montrer comment détecter ces objets à travers des méthodes d'analyse et de segmentation d'images. Ensuite nous allons montrer dans le chapitre 4 une technique de suivi de ce type d'objet à travers des techniques de segmentation spatio-temporelle.

Chapitre 3

Analyse d'image pour une représentation pseudo sémantique

Les systèmes de vision par ordinateur travaillent dans la plupart des cas sur une représentation de l'image/vidéo simple et exploitable, plus facile à interpréter qu'un ensemble de pixels. Dans le cas où un robot évolue dans un environnement intérieur et dynamique, différentes entités vont apparaître dans son entourage de type statique (meuble, porte) ou dynamique (personnes, fauteuil). Dans ce contexte, un système de navigation doit être capable d'extraire l'ensemble des objets d'intérêts qui composent la scène. De ces informations, il s'agira à terme de planifier des actions pour assurer la sécurité durant la navigation du fauteuil en détectant et évitant les obstacles.

Ce type d'extraction requiert une représentation de haut niveau conforme au contenu sémantique de la scène. Toutefois, avant d'identifier les objets dans une image, il faut au préalable extraire leur forme ou leur couleur. Dans notre cas, on ne dispose d'aucun a priori sur les objets qui peuvent apparaître dans la scène. Ainsi, au lieu de fournir une représentation en objets, on se limite à une représentation en régions pseudo sémantique qui correspond à des surfaces homogènes dans l'image [Babel 12]. Cette représentation peut être une étape précédant l'identification des objets. Cette extraction de régions est assurée par une technique de segmentation dédiée qui consiste à rassembler les pixels/blocs de l'image en fonction de certains critères de similarité ou de dissimilarité. Les critères de similarité (ou critères d'homogénéité) peuvent être définis suivant différents types de descripteurs (couleur, texture, gradient).

La segmentation est un processus fondamental dans l'analyse d'image. Elle est généralement utilisée comme prétraitement dans différents types d'applications (médicale, surveillance, robotique etc.). L'objectif de la segmentation est de passer

Figure 3.1 – Segmentation d'image naturelle

d'une représentation de l'image en pixels à une représentation pseudo sémantique basée régions (cf. figure 3.1).

Dans le cadre de nos travaux, nous disposons d'une caméra couleur embarquée sur le fauteuil. De ce fait, nous allons développer une technique de segmentation adaptée. Le but sera de fournir une représentation pseudo sémantique afin de réduire la quantité d'informations interprétables par le système au niveau région.

Dans ce chapitre, nous présentons une nouvelle technique de segmentation nommée JHMS (Joint Hierarchical and Multiresolution Segmentation) [Sekkal 12]. L'originalité de cette approche réside dans ses propriétés multi-échelle. L'image est alors substituée par une représentation intermédiaire offrant à la fois les propriétés de multirésolution et de compacité.

Le chapitre est divisé en quatre sections : tout d'abord, une courte introduction sur les bases d'une représentation pseudo sémantique est présentée dans la section 3.1. Ensuite, la section 3.2 présente quelques techniques de segmentation des images couleur de l'état de l'art. La section 3.3 aborde le schéma général de la technique de segmentation proposée ainsi que l'algorithme et le choix des descripteurs nécessaires. Enfin, les expérimentations et les résultats sont présentés dans la section 3.4.

3.1 Représentation pseudo-sémantique de l'image

Le contenu de l'image peut être représenté par différents niveaux de signification sémantique. Les techniques visant à extraire des objets comme une entité sémantique fournissent une représentation de haut niveau de la sémantique. Cependant, des outils intermédiaires permettent d'obtenir une représentation cohérente en termes de couleur, texture... Cette représentation intermédiaire peut être qualifiée de représentation pseudo-sémantique puisque elle s'adapte au contenu de l'image [Babel 12].

Une représentation pseudo-sémantique est donc composée d'un ensemble de régions homogènes en termes de différents descripteurs issus d'un processus de segmentation d'image. Une région est définie dans [Castagno 98] comme étant une surface dans une image qui respecte une certaine homogénéité selon un critère défini au préalable. Un objet est quand à lui caractérisé par un contenu sémantique, même s'il peut être composé de plusieurs régions incohérentes entre elles en termes de couleur, texture ou de mouvement.

Les régions doivent respecter un ensemble de critères pour représenter le contenu de l'image. Dans [Horowitz 76], les auteurs ont formulé le problème de la segmentation comme suit. Considérons

- Π un ensemble de régions R de l'image I,
- R_i une région de la partition Π,
- N le nombre de régions de la partition Π,
- $P(R_i)$ un prédicat qui permet de vérifier l'homogénéité de la région R_i.

Une partition Π doit vérifier les propriétés suivantes :

1. $\Pi = \cup R_i$, $i \in \{1...N\}$: une partition Π doit être complète
2. $\forall (i,j)\ R_i \cap R_j = \emptyset$: les régions doivent être disjointes
3. $\forall i\ R_i \neq \emptyset$
4. $\forall i\ P(R_i) =$ vrai : toutes régions doivent satisfaire le critère de similarité
5. $\forall i\ P(R_i \cup R_j) =$ faux si R_i et R_j sont adjacentes. Ainsi, il s'agit de s'assurer qu'il n'existe pas deux régions adjacentes dont l'union peut satisfaire le critère de similarité.

Maintenant que les bases d'une représentation pseudo-sémantique sont posées, nous allons présenter un aperçu des techniques de segmentation se trouvant dans l'état de l'art.

3.2 Techniques de segmentation

Le problème de la segmentation d'image a été abordé dans de nombreux travaux. Parmi les premiers chercheurs qui ont étudié la segmentation on peut citer Horowitz et Palvadis [Horowitz 76], en proposant le premier algorithme de segmentation d'image. Depuis, des améliorations ont été apportées à sa technique et plusieurs approches qui reposent sur des principes différents ont vu le jour. Il existe de nombreuses méthodes de segmentation qu'on peut regrouper en deux classes principales :

– segmentation basée homogénéité,
– segmentation basée discontinuité.

Le but de cette section est donc de décrire un ensemble non exhaustif de techniques de segmentation d'image existantes dans l'état de l'art. Les techniques présentées vont permettre de justifier notre choix de segmentation tant par l'approche de segmentation que par les propriétés qu'elles présentent.

3.2.1 Segmentation basée détection régions homogènes

Cette première famille d'approches consiste à rechercher les régions homogènes dans l'image. La segmentation est basée sur un critère d'homogénéité (niveaux de gris, couleur, texture). Dans la suite, nous allons présenter quelques techniques de segmentation basée sur la détection de région homogènes.

3.2.1.1 Seuillage d'histogrammes

Les techniques de segmentation par seuillage reposent sur l'hypothèse que les régions appartenant à la scène ont des distributions de niveaux de gris différentes permettant de les discriminer [Albuquerque 04]. Pour cela, ces techniques se basent sur l'observation de l'histogramme qui contiendra des pics correspondant aux objets. La segmentation consiste à définir des seuils de séparation et à attribuer la même étiquette à tous les pixels se trouvant dans le même intervalle de niveaux de gris. La détection des seuils de séparation à partir des histogrammes consiste à trouver un ensemble de valeurs permettant d'extraire plusieurs modes (pics) dans l'histogramme. Dans [Albuquerque 04], la fonction d'entropie est utilisée pour estimer le seuil. [Guo 03] utilise une fonction basé sur le critère Fisher sur l'ensemble des valeurs de l'histogramme afin de déterminer le seuil de découpage. Cette technique très simpliste ne peut s'appliquer aux images naturelles.

3.2.1.2 Croissance de régions

Une approche dite par croissance de régions (Region Growing Algorithm) consiste à faire agglomérer progressivement les pixels de l'image autour d'un point de départ appelé généralement *germe* pour former des régions homogènes. Les germes sont initialement fixés dans des endroits spécifiques de l'image. Dans [Zucker 76], les auteurs proposent un algorithme simple et très répandu se basant sur la technique de croissance de régions. Le processus se compose de deux étapes importantes :

1. **Sélection des germes :** Le choix des germes est une étape importante pour le bon déroulement du processus de segmentation. En effet, le regroupement des pixels autour de ces points va s'effectuer en se basant sur des mesures de similarité estimées autour du point choisi. Il faut donc que les germes soient localisés dans des zones à faible variation d'énergie (région homogène). Les germes doivent respecter les critères suivants [Shih 05] :
   - un germe doit avoir une forte similarité avec ses voisins,
   - pour extraire un région, au moins un germe doit être défini,
   - aucun germe de régions différentes ne doit être connexe.

   Dans ce cas, le processus pourra regrouper les pixels du voisinage jusqu'à atteindre une rupture (contours). Si au contraire, les germes sont sélectionnés sur une zone non homogène, le critère de similarité sera biaisé par les variations locales ce qui donnera de mauvais résultats.

2. **Croissance des régions** : Cette étape a pour but de regrouper les pixels se trouvant sur le voisinage tout en essayant de maintenir l'homogénéité de la région. Un pixel est donc inclus dans une région existante si le critère est vérifié ou inclus dans une nouvelle région sinon. Le processus de croissance s'arrête lorsque plus aucun pixel voisin ne peut être ajouté sans modifier l'homogénéité de la région.

Différents travaux ont été réalisés en se basant sur l'approche de croissance de régions. [KaraFalah 94] autorise la fusion des pixels qui possèdent un faible gradient à chaque itération du processus. Une autre stratégie décrite dans [Gambotto 93] combine la détection des contours pour guider la croissance des régions. La fusion des pixels est contrôlée par les contours détectés pour stopper la croissance des régions. Le critère d'homogénéité proposé dans [Xiaohan 92] consiste en une somme pondérée du contraste entre la région et le pixel. De ce fait, plus le résultat est faible plus le pixel est susceptible d'appartenir à la région.

Plus récemment, plusieurs améliorations on été proposées tant au niveau de la sélection des germes qu'au niveau de la stratégie de fusion. Les auteurs dans [Cui 08] proposent d'adapter le seuil de fusion entre deux régions en utilisant un modèle d'incertitude autour de chaque germe sélectionné. Dans [Preetha 12, Shih 05], la sélection des germes se fait en fonction de la distance Euclidienne entre l'intensité d'un pixel et ses voisins. [Preetha 12] décide de la fusion des régions en fonction de leurs tailles et une fonction d'homogénéité. [Tang 10] propose une technique de segmentation d'image couleur en se basant sur une sélection de germes obtenue par une technique de partage des eaux.

Le problème du schéma de segmentation basé croissance de région, est que à chaque itération, une ou plusieurs fusions sont effectuées. Le résultat de la segmentation dépend donc fortement de l'ordre dans lequel les fusions ont eu lieu.

3.2.1.3 Division-Fusion

La technique de segmentation division-fusion se base sur le principe suivant : à partir d'une seule région représentant l'image, on applique une opération de division récursive tant que l'homogénéité n'est pas respectée. Ensuite, on réobserve l'homogénéité des régions adjacentes pour les fusionner éventuellement.

Considérons $P_1(R_i)$ et $P_2(R_i)$ comme des fonctions qui permettent de vérifier respectivement les critères d'homogénéité des étapes de division et de fusion. Formellement, le processus de division/fusion se résume à :

1. décomposer chaque région R_i dont le prédicat $P_1(R_i) = faux$
2. fusionner toutes les régions adjacentes R_i et R_j si $P_2(R_i \cap R_j) = vrai$

Le processus de division consiste à faire une décomposition de l'image en petites régions de tailles différentes. Dans la plupart des travaux, la division consiste en une décomposition par blocs de l'image, conduisant à une représentation appelée *quadtree* [Grosky 83]. Cette représentation est une structure de données de type arbre (cf. figure 3.2). La racine de l'arbre correspond à l'image entière et chaque noeud dans l'arbre contient exactement quatre feuilles. Les feuilles correspondent aux régions répondant au critère d'homogénéité. Le processus de décomposition est répété jusqu'à ce qu'il n'y ait plus aucun bloc qui nécessite d'être décomposé ou que les blocs sont de la taille d'un pixel. La division en bloc permet de réduire la quantité d'information à traiter lors de l'étape de fusion en passant d'une représentation du niveau pixel à un niveau bloc plus compacte et qui conserve les mêmes caractéristiques.

Les premiers travaux sur la segmentation par division/fusion sont très anciens. Horowiz et Pavalivis [Horowitz 74] ont été les premiers à proposer un algorithme de segmentation de ce type. [Ohlander 78] décompose d'abord l'image en régions en fonction de leur histogramme en niveaux de gris. Dans [Montoya 03], les auteurs proposent le regroupement des pixels comme pour le processus de croissance des régions. Les points de départ seront sélectionnés sur des blocs de grandes tailles puisqu'ils sont caractérisés par une faible variation de luminance.

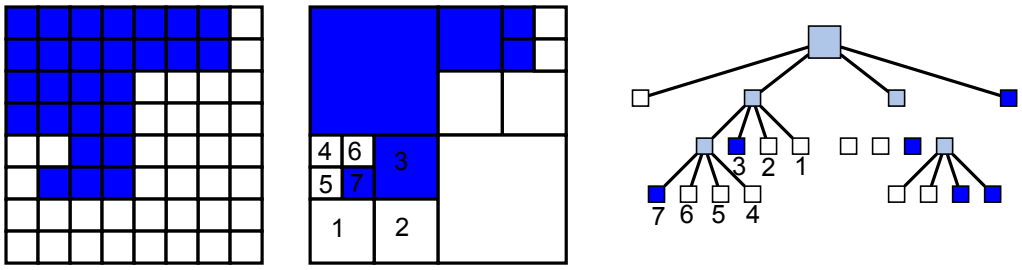


Figure 3.2 – Exemple de décomposition de quadtree

Plusieurs améliorations ont été apportées dans la dernière décennie. Par exemple les travaux [Merigot 03, Aneja 09] se basent sur une nouvelle technique de décomposition appelé *décomposition optimale* qui permet de réduire le nombre de régions homogènes. Au lieu de décomposer la région en quatre sous régions de même taille (décomposition quadtree), la technique consiste à découper la région en deux sous régions en cherchant la coupe qui maximise la dissimilarité entre les deux régions (maximum de la différence d'intensité moyenne des deux régions). De nouvelles techniques de segmentation division/fusion basées sur des modèles probabilistes ont vu le jour. Dans [Zhang 03, Li 09], les auteurs utilisent l'algorithme d'espérance maximisation pour le processus de division et fusion.

3.2.1.4 Classification

Appelées aussi *clustering*, les méthodes de classification représentent une autre manière de détecter des régions homogènes (clusters). La région est définie par un ensemble de caractéristiques définies dans un espace de N dimensions (couleur/espace). Partant de l'hypothèse que les pixels de la même région doivent être spatialement proches et similaires en termes de couleur afin de former les clusters, l'ensemble des pixels appartenant au même cluster peut alors former une distribution caractéristique (typiquement normale) ce qui facilitera les décisions d'appartenance entre classes [Celenk 90].

Dans cette catégorie d'algorithme, la technique des K-moyennes est très populaire [Hartigan 79]. Cette technique est basée sur la minimisation de l'indice de performance qui est calculé par la somme de la distance euclidienne de chaque pixel avec les centres de chaque cluster. Au départ, une première partition est fournie assignant chaque pixel au cluster qui minimise la distance euclidienne. Ensuite, les pixels sont réaffectés à d'autres clusters jusqu'à converger vers un minimum de l'indice de performance [Gevers 90].

[Oliver 06] propose une amélioration de la technique de classification en procédant en deux étapes : une première étape consiste à classifier tous les pixels sauf ceux qui ont un fort gradient (contours) et ceux qui sont à une distance importante du cluster (bruit). Ensuite, un raffinement des contours est effectué sur les pixels non classés.

La technique de Mean Shift est un algorithme non paramétrique de partitionnement de données multidimensionnelles [Comaniciu 02]. Cette technique repose sur un processus itératif qui consiste à associer à chaque pixel la moyenne de son voisinage en utilisant un noyau (exemple : noyau gaussien). Ensuite, les pixels associés à la même moyenne sont fusionnés ensemble formant des régions homogènes (cluster). Cette technique reste cependant très coûteuse et dépend de la taille de la fenêtre de voisinage (plus la fenêtre est grande, plus la segmentation est lente).

Le problème de la classification est qu'il faut connaître au départ le nombre de régions à extraire. Ceci implique une interaction humaine afin de spécifier le nombre de régions ce qui limite son utilisation pour des applications automatiques. Cependant, certaines techniques tentent de pallier ce problème en initialisant un nombre important de clusters pour produire une sur-segmentation. Ensuite, une étape de fusion de régions est appliquée afin de réduire le nombre de régions [Ursani 08].

Les techniques de classification suivent une approche globale pour la segmentation d'image. Elles souffrent d'un manque d'informations locales lors du regroupement des pixels ce qui peut engendrer une sur-segmentation en régions (cas d'un dégradé de couleur). Par contre, le processus de division-fusion repose sur une approche combinant une étape de division globale pour décomposer l'image en régions similaires et une deuxième étape de fusion locale pour regrouper les régions similaires. Cette approche va permettre une extraction fidèle au contenu de l'image.

3.2.2 Segmentation par détection de discontinuités

Dans cette deuxième famille d'approches, la segmentation se fait par rapport à une recherche des frontières des régions. Ces frontières sont caractérisées par une forte variation locale d'énergie. L'image de gradient est typiquement utilisée pour mesurer la discontinuité du signal à l'aide d'un seuil. La détection des contours est donc l'idée principale de la segmentation par détection de discontinuités.

3.2.2.1 Détection de contours

Les contours sont essentiels dans la perception des objets par le système visuel humain. La détection des contours consiste à localiser les changements abrupts entre les points voisins des pixels par un seuil sur le gradient éventuellement combiné avec un test de passage par zéros du Laplacien. La notion de contour étant reliée à celle de la variation, il est donc naturel de baser les détecteurs de contours sur des méthodes dérivatives : une variation existera si le gradient est localement maximal ou si la dérivée seconde présente un passage par zéro. Canny [Canny 86], Sobel [Duda 73] ou Robert [Roberts 63] sont les premiers filtres utilisés pour la détection des contours. Plus récemment, [Martin 04] définit un opérateur de gradient adapté à la luminance, la couleur et la texture pour l'utiliser dans un classifieur afin de prédire la force d'un contour. L'apprentissage se fait sur un ensemble d'images segmentées à main levée. Dans les cas d'une image couleur, un filtre dérivatif est appliqué à chaque composante couleur de l'image. Ensuite, les images de gradient sont combinées pour extraire les contours.

Le plus gros inconvénient des techniques basées contours est la non fermeture des contours. En fait, une représentation en région est constituée par un ensemble de régions délimitées par des contours fermés. Ceci n'est plus vrai lorsqu'on cherche à détecter des contours sur la base des dessimilarités locales. Pour cela, il faudra ajouter une étape appelée fermeture de contours [Elder 96, Jiang 00, Arbelaez 11] qui consiste à relier les extrémités des contours qui sont supposés appartenir à la même région.
La technique de [Maire 08] permet d'extraire des régions à partir d'une carte de contours en se basant sur une extension de la technique des partage des eaux. Le lecteur peut se référer à [Mittal 12] pour plus de détails sur les approches de détection de contours.

3.2.2.2 Contours actifs

Une autre approche très populaire est souvent présentée sous le nom de contours actifs. Les contours actifs ont, pour la première fois, été proposés par Kass [Kass 88]. Par la suite, un nombre de variantes ont été proposées [Gosda 10] utilisant par exemple un filtre de Canny pour déterminer les contours dans l'image. [Herbulot 07] présente dans sa thèse un état de l'art plus complet sur les techniques de segmentation à base de contours actifs. Quelques difficultés se posent avec ce type d'approche : généralement, cette technique est utilisée pour extraire un objet d'intérêt dans une image. Cette extraction repose sur une l'initialisation manuelle ainsi que des problèmes de convergence du contour sur l'objet à segmenter.

3.2.3 Segmentation scalable

La scalabilité est une propriété intéressante dans le processus de segmentation. Elle permet de gérer à la fois les propriétés locales et globales au niveau de l'image et des régions.
Deux concepts de scalabilité ont été introduits dans la littérature. Le premier est la scalabilité spatiale (cf. figure 3.3(a)) ou plus communément appelée multirésolution. Le deuxième concept est la scalabilité sémantique ou appelée hiérarchie (cf. figure 3.3(b)).

La multirésolution consiste à représenter une image en un ensemble d'images de différentes résolutions constituant une pyramide d'images. Dans ce type de segmentation, le processus consiste généralement en une approche descendante. L'idée est de traiter d'abord l'image de basse résolution avec le minimum d'information, ceci dans l'objectif de fournir une première segmentation grossière. Ensuite, la technique consiste à raffiner la segmentation en injectant de l'information tirée des niveaux inférieurs de la pyramide.

La hiérarchie consiste à fournir des représentations de la segmentation à différents niveaux de sémantique. Partant d'une sur-segmentation avec des régions de petites tailles, la segmentation est modifiée en réduisant le nombre de régions jusqu'à atteindre des régions qui peuvent décrire des objets sémantiques dans la scène.

Que ce soit la multirésolution ou la hiérarchie, la scalabilité est souvent représentée par une structure pyramidale permettant la représentation et le traitement de l'image. Il existe deux types de pyramides : *les pyramides régulières* pour la segmentation multirésolution et *les pyramides irrégulières* pour la segmentation hiérarchique. Dans la section suivante, nous allons détailler les deux structures pyramidales.

3.2.3.1 Pyramides régulières

Les pyramides régulières représentent un ensemble d'images à différents niveaux de résolution obtenues selon un processus itératif de filtrage et de sous-échantillonnage (cf. figure 3.3(a)). Le premier niveau de la pyramide consiste en une image de pleine résolution. Ensuite pour chaque niveau, la résolution est réduite d'un facteur de réduction constant entre les niveaux.

Une pyramide régulière est définie par un ratio $\frac{N \times N}{q}$ avec $N \times N$ qui correspond à la taille de la fenêtre de réduction et q le facteur de réduction. Si le ratio $\frac{N \times N}{q}$ est supérieur à 1, cela signifie que les fenêtres de réduction sont en

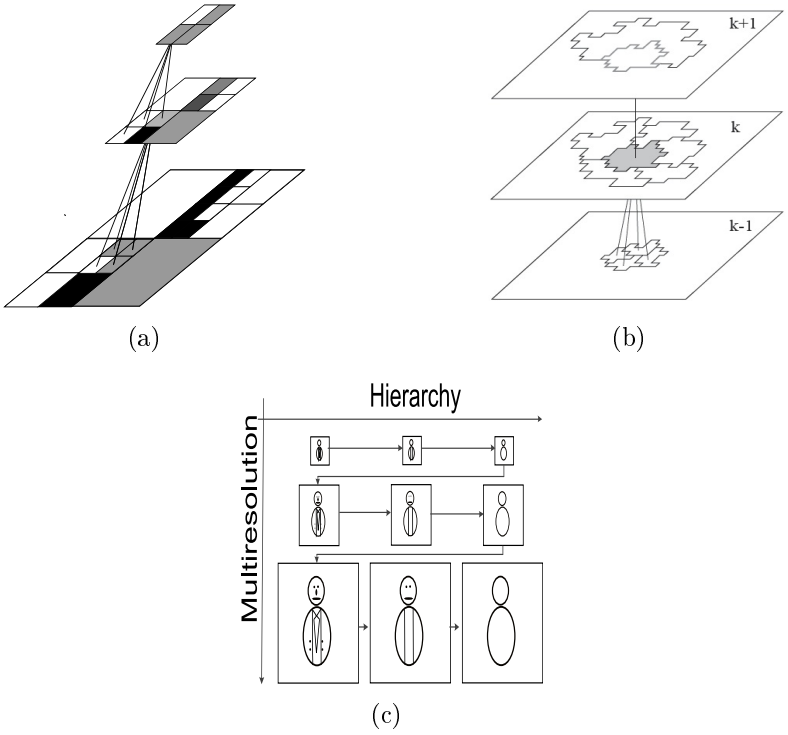


Figure 3.3 – Segmentation hierarchique et multirésolution

recouvrement. Ceci permet de définir la structure rigide des pyramides régulières [Brun 02].

Une des caractéristiques importantes des pyramides régulières est la bonne conservation des relations intra et inter niveaux dans la pyramide. En effet, un élément (pixel) dans la pyramide est dit voisin d'un autre s'ils sont adjacents dans le même niveau de pyramide. D'autre part, chaque pixel d'un niveau $l+1$ possède $N \times N$ fils dans le niveau inférieur l. Cette caractéristique permet de simplifier l'accès aux éléments dans la pyramide.

Différents types de structures pyramidales ont été proposés. Tout d'abord, les plus simples sont les pyramides gaussiennes présentées dans [Burt 83] qui consistent à réduire la résolution de l'image d'un facteur égal à 4 pour chaque niveau de la pyramide en appliquant un filtrage passe bas et un sous-échantillonnage de la taille d'image par 4. Les pyramides laplaciennes [Burt 83] quant à elles, donnent une décomposition fréquentielle de l'image en mettant les composantes de plus basses fréquences en haut de la pyramide.

Chen et Pavlidis [Chen 80] ont proposé la première technique de segmentation pyramidale. Ils définissent pour cela deux types de relations entre les éléments de la pyramide : *verticales* pour les relations père-fils entre deux niveaux successifs de la pyramide et *horizontales* pour définir les relations frère-frère entre les éléments du même niveau.

Plus récemment, [Stojmenovic 10] propose une technique de segmentation combinant les descripteurs couleur et texture. Il utilise deux pyramides : une première pyramide pour la couleur, et une deuxième pyramide laplacienne pour décrire la texture de l'image. Ici la similarité entre deux régions est estimée en fonction de la différence des couleurs et de l'orientation du gradient.

Dans [Vantaram 09a], les auteurs présentent une technique de segmentation multirésolution basée sur un critère de gradient adapté au niveau de résolution. En effet, ils supposent que le gradient dans l'image à basse résolution est plus faible que le gradient se trouvant dans les images de pleine résolution. La segmentation multirésolution réside sur le principe de fusion père/fils : si les descripteurs de couleur du pixel fils sont comparables à ceux de son père alors le pixel fils hérite de la même étiquette que son père.

Les pyramides régulières présentent plusieurs propriétés présentées dans [Bister 90]. La première consiste en la réduction du bruit : en construisant la pyramide, les détails non significatifs qui peuvent fausser la segmentation sont éliminés. Aussi, une représentation à différents niveaux de résolution spatiale permet de passer des observations globales aux observations locales. De plus, la réduction de la taille de l'image tout en conservant les relations spatiales permet de diminuer le temps de calcul. Enfin, ce type d'approche offre une meilleure adaptabilité de l'ensemble des paramètres qui contrôlent le processus de segmentation en fonction du niveau de la pyramide sur lequel les paramètres sont appliqués.

Cependant, le modèle pyramidal régulier présente quelques faiblesses dues à la rigidité de la structure comme la variation à la translation et l'apparition d'artefacts dans les régions [Gross 87], et aussi la difficulté de détecter les objets allongés. Le lecteur pourra se référer à [Braviano 95, Marfil 06] pour plus d'informations sur les approches pyramidales.

3.2.3.2 Pyramides irrégulières

Les premiers travaux sur les pyramides irrégulières ont essayé de maintenir un facteur de décimation fixe entre les niveaux. L'idée étant que sur une machine

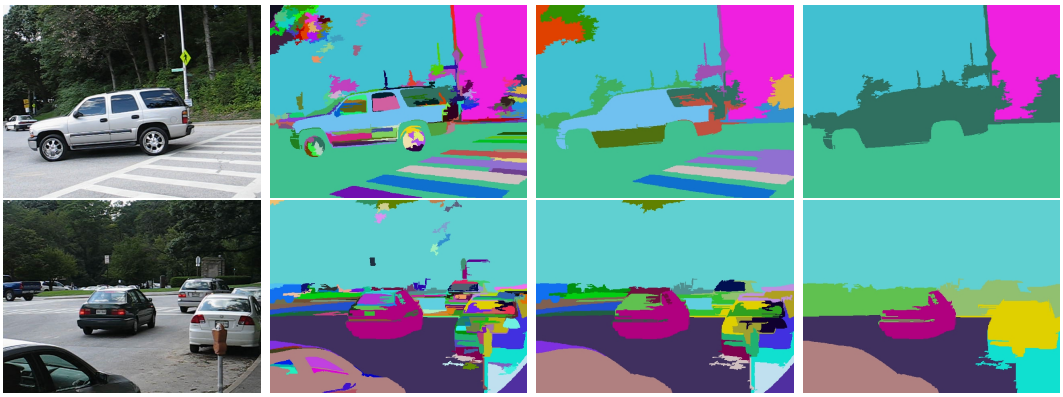


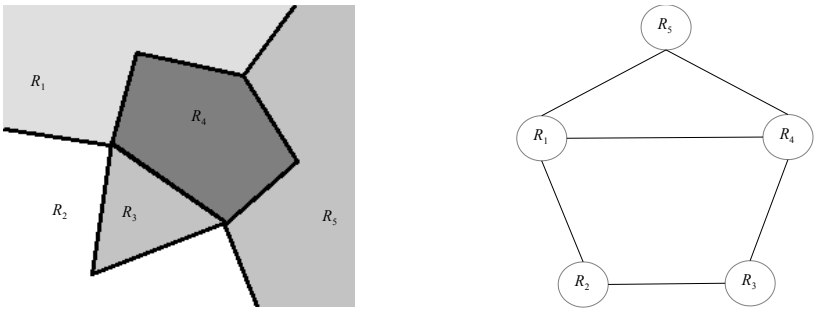
Figure 3.4 – Segmentation à différents niveaux de hiérarchie

idéale complètement parallèle l'obtention de chaque niveau peut se faire en temps constant. Donc maintenir un facteur de décimation fixe permet (en théorie) d'avoir des algorithmes parallèles fonctionnant en log() de la taille de l'image. Les pyramides irrégulières permettant de garantir la connexité des régions et leurs relations d'adjacence contrairement aux pyramides régulières.

Appelées aussi pyramides de graphes, les pyramides irrégulières permettent de fournir une segmentation à différentes échelles en un seul traitement. Ce type d'approches repose sur la construction d'un graphe d'adjacence. En effet, une segmentation peut être représentée par un graphe d'adjacence non orienté RAG (Region Adjency Graph) [Colantoni 97]. Les sommets du graphe correspondent aux régions et les arêtes indiquent une connexité des deux régions qu'ils relient (cf. figure 3.5). Cette représentation permet de fournir l'information d'adjacence de la carte de segmentation. Les liens dans le graphe contiennent des informations de ressemblance entre deux régions voisines. Ceci permet en particulier la fusion des régions et la simplification du graphe. L'un des avantages du graphe d'adjacence de régions est qu'il permet de fournir une vue spatiale de l'image [Schettini 93].

Initialement, au premier niveau de la pyramide, chaque pixel est lié à un sommet du graphe. Ensuite, en se basant sur des descripteurs locaux appliqués à chaque sommet et à ses voisins, des opérations de fusion sont effectuées afin de regrouper les sommets qui respectent les critères de similarité.

L'opération de réduction d'un graphe a été introduite pour la première fois par Meer [Meer 89]. D'un niveau à un autre, le nombre de sommets dans le graphe va diminuer suite à leur fusion. A chaque niveau l de la pyramide, un graphe $G_l = (V_l, E_l)$ est calculé où V_l est l'ensemble des sommets au niveau l



(a) (b)

Figure 3.5 – Représentation en RAG : (a) segmentation d'images contenant 5 régions R_i avec $i = 1 \ldots 5$ (b) le RAG correspondant

et E_l est l'ensemble des arêtes entre les sommets. Le processus de segmentation identifie à chaque itération un sous-ensemble de V_l pour construire le prochain graphe G_{l+1}. Les sommets sélectionnés sont appelés les survivants. Le reste des sommets du graphe G_l sont tous fusionnés avec les survivants en fonction de critères de similarités.

Donc, pour chaque sommet du graphe G_{l+1}, on associe un ensemble connexe de sommets du graphe précèdent G_l. Cela permet de définir les relations père-fils dans la pyramide ainsi que le facteur de réduction entre deux niveaux successifs qui varie d'un sommet à un autre.

Pour résumer, à chaque itération, le processus de segmentation consiste à

1. sélectionner les survivants du niveau l pour construire V_{l+1},
2. attribuer chaque non-survivant du niveau l à un survivant pour créer les relations père-fils,
3. enfin, mettre à jour les relations de voisinage du graphe G_{l+1}.

Une des premières approches utilisant les pyramides irrégulières a été proposée dans [Meer 89]. La segmentation est réalisée d'une manière ascendante en appliquant la sélection des sommets survivants présentée dans le paragraphe précèdent. [Bertolino 96] propose d'adapter les paramètres de fusion (moyenne, variance) pour chaque niveau de pyramide. Dans [Cheng 00], les auteurs présentent une technique de segmentation hiérarchique : les régions homogènes sont d'abord identifiées à travers une technique de seuillage multi-niveaux. Ensuite, les régions sont représentées dans l'espace couleur CIE(L*a*b) et fusionnées si elles respectent un seuil calculé adapté à chaque niveau de hiérarchie.

Une autre approche hiérarchique basée cette fois sur une extraction de partition à partir d'images de contour est proposée dans [Arbelaez 11]. Pour ce faire, les auteurs proposent une extension de l'algorithme de partage des eaux afin de construire un ensemble de régions à partir de l'image des contours. A partir de cette représentation, les contours des régions sont pondérés suivant si ils sont issus de la carte des contours ou si les régions obtenues par la technique de partage des eaux. Ensuite, les régions adjacentes sont fusionnées en fonction du poids accordé au contour les séparant. Afin de pouvoir réaliser une segmentation hiérarchique, la carte des contours doit contenir des contours pondérés suivant le niveau des régions qu'ils séparent : deux objets sémantiques doivent être séparés par un contour plus important que ceux issus d'une simple texture.

3.2.4 Discussion

Dans cette section, nous avons présenté un ensemble non exhaustif des techniques de segmentations d'images. Nous avons montré que pour avoir une représentation en régions correcte (avec des contours fermés), il est plus intéressant d'opter pour les techniques basées sur l'homogénéité (croissance de région ou division/fusion). De plus, afin de réduire la complexité du calcul, on a vu que l'utilisation des pyramides régulières permet de fournir une représentation compacte de l'image. D'un autre côté, les techniques de segmentation hiérarchique fournissent un résultat à différents niveaux sémantiques de la scène observée. Ceci offre la possibilité à l'utilisateur de sélectionner le niveau de sémantique adapté au type d'application souhaitée.

Pour cela, nous allons proposer une technique de segmentation qui permet de combiner à la fois la multirésolution (pour réduire la complexité) et la hiérarchie (pour un résultat riche en sémantique). Nous basons notre technique de segmentation sur la division/fusion puisque une représentation quadtree est conforme avec le contenu de l'image et facile à calculer et reste surtout compatible avec une représentation multirésolution de l'image.

Dans la section suivante nous présentons cette technique de segmentation développée au cours de ces travaux de thèse appelée JHMS (*Joint Hierarchical and Multiresolution Segmentation*) qui permet de répondre aux contraintes de scalabilité.

3.3 Contribution : JHMS

Dans l'algorithme de segmentation proposé (JHMS), nous combinons à la fois les propriétés de scalabilité spatiale (multirésolution) et sémantique

(hiérarchique) afin de réaliser une représentation pseudo-sémantique efficace de l'image. A cet effet, nous exploitons des représentations multi-échelles afin de tirer avantage des descripteurs d'image à différents niveaux de résolution. Cette représentation permet aussi de réduire la complexité de calcul puisqu'elle permet de détecter des régions dans des basses résolutions de l'image. D'autre part, afin d'assurer une représentation en différents niveaux sémantiques, l'algorithme fournit des représentations avec plusieurs niveaux de granularité. Cette technique sera basée sur un regroupement des régions en fonction de leurs similarités.

La figure 3.6 présente le schéma bloc général de la segmentation proposée. Deux blocs fonctionnels principaux sont indiqués : la partie multirésolution représentée par la gestion de la pyramide et le quadtree, et la partie hiérarchique représentée par l'algorithme de fusion exécuté à chaque niveau de résolution.

La suite de cette section s'attache à décrire en détail les étapes de calcul de l'algorithme JHMS. Ce travail a fait l'objet d'une publication dans [Sekkal 12].

3.3.1 Représentation multirésolution

Avant de se lancer dans la description détaillée de la technique, nous allons présenter les outils utilisés pour la représentation de l'image, notamment la décomposition quadtree nécessaire pour la technique de division/fusion. Ensuite, nous présentons la construction de la pyramide multirésolution de l'image.

Extraction du quadtree

La segmentation proposée dans ce chapitre repose sur une représentation intermédiaire quadtree [Grosky 83]. Ce choix est justifié par la fidélité de cette représentation par rapport à l'image originale et son coût de calcul quasi négligeable par rapport au processus de segmentation. Dans notre approche, on calcule une collection de quadtrees au lieu d'utiliser l'approche classique qui consiste à considérer l'image entière et la décomposer en plusieurs blocs homogènes [Pasteau 10].

Tout d'abord, on décompose l'image en une grille de blocs de taille $N_{max} \times N_{max}$. Ensuite, pour chaque bloc, un quadtree est calculé en subdivisant les blocs qui contiennent un fort gradient en quatre sous blocs. Cette opération est répétée jusqu'à ce qu'il n'existe aucune autre division possible ou que les blocs sont de la taille d'un pixel. Techniquement, $N_{max} = 2^L$ pour une division régulière, avec L le nombre de niveaux dans la pyramide. En termes de résultats, on obtient une image quadtree I_q contenant des blocs de différentes tailles $N \times N$

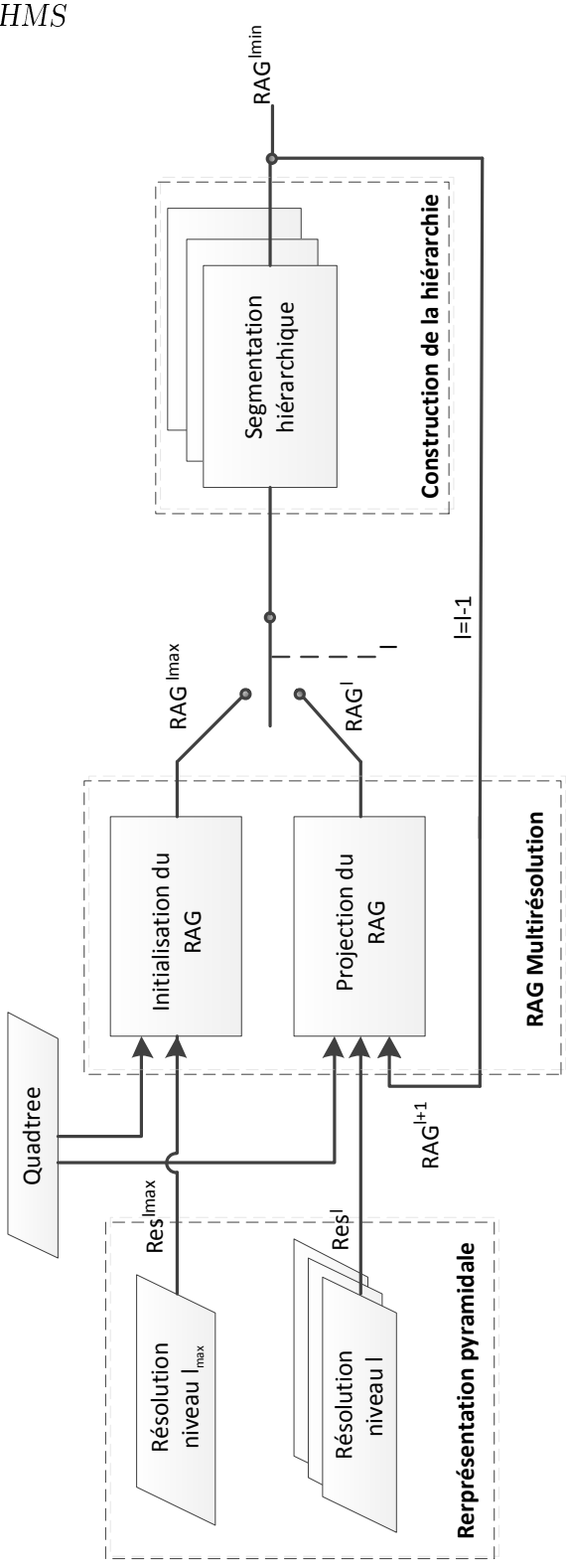


FIGURE 3.6 – Schéma général de la segmentation JHMS

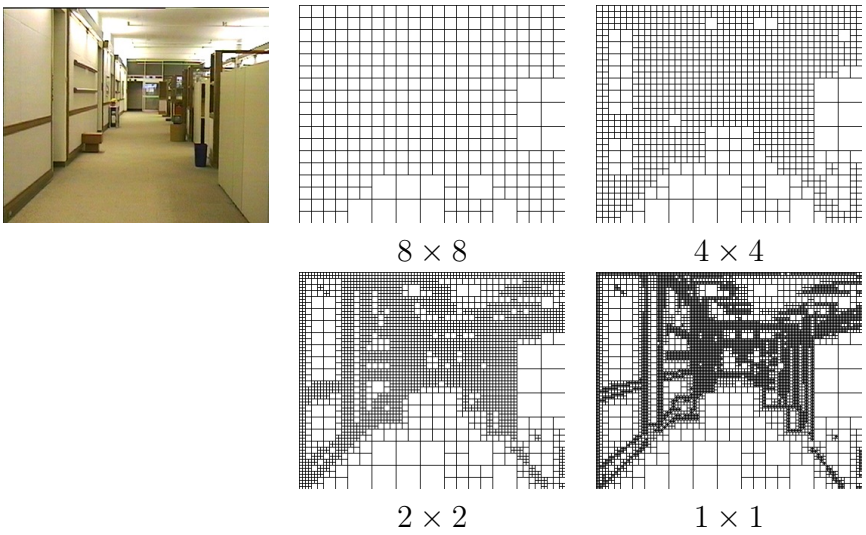


Figure 3.7 – Construction du quadtree sur la 1ère image de la séquence hall en résolution cif (352x288) en utilisant 5 niveaux de blocs

avec $N = \{1, \ldots, N_{max}\}$.

La décision de division ou non d'un bloc en quatre est réalisée en fonction de la variation d'énergie se trouvant dans ce dernier. Cette variation est estimée en fonction de la distance entre les valeurs minimale et maximale de la luminance des pixels qui composent le bloc (gradient). Un seuil T_y est fixé empiriquement pour décomposer tout bloc ayant ce gradient supérieur à T_y en quatre sous blocs.

Dans la figure 3.7, les différentes représentations quadtree lors des étapes de division sont présentées. De gauche à droite, on peut voir l'évolution de la construction du quadtree : les blocs se trouvant sur les régions à fort gradient (typiquement les contours) dans l'image originale sont successivement divisés.

Construction de la pyramide régulière

Les niveaux de la pyramide sont construits d'une manière itérative en réduisant à chaque niveau la résolution par un facteur quatre. Les fenêtres de réduction ne sont pas en recouvrement. La base de la pyramide correspond à l'image de texture du quadtree. Ainsi, pour chaque bloc dans le quadtree, la moyenne de sa luminance/couleur est calculée et placée dans le quadtree. Ceci permet de réduire drastiquement la quantité d'informations traitées à ce niveau de la pyramide qui correspond à la pleine résolution [Déforges 07]. A ce niveau de traitement, se pose le premier problème : comment fixer le seuil T_y pour la décomposition quadtree. Diminuer le seuil T_y permet de préserver les contours se trouvant sur les régions

à gradient moyen mais le nombre de blocs du quadtree sera alors très important. Par contre, augmenter le seuil T_y réduit le nombre de blocs dans le quadtree en causant une perte significative dans les détails au niveau des contours. Pour cela, un compromis doit être fait pour le choix du seuil qui concilie à la fois la préservation des détails et la réduction de la quantité d'information.

3.3.2 RAG multirésolution

Le processus de segmentation produit à chaque niveau de pyramide un RAG comme résultat. Ce RAG définit les régions, leurs caractéristiques et leurs relations de voisinage. Cette structure permet aussi de réaliser une segmentation hiérarchique dans le but de produire une pyramide irrégulière à chaque niveau de la pyramide.

Au départ, le RAG est initialisé à partir de l'image basse résolution de la pyramide. Les sommets dans le RAG vont représenter les pixels dans cette image, et les relations d'adjacence correspondent au relations de voisinage à 4 entre les pixels.

Le processus effectue d'abord une segmentation grossière dans le haut de la pyramide produisant le graphe RAG^{lmax}. Ensuite, cette segmentation est raffinée à chaque niveau de la pyramide. Pour cela, nous proposons une technique de projection du RAG d'un niveau à un autre afin de réutiliser et exploiter à bien les résultats d'une segmentation pour la raffiner. On introduit donc dans cette section le RAG multiresolution, une structure de graphe capable de se projeter d'un niveau à un autre, en restant flexible à différents types d'opérations que l'on va présenter juste après.

Les étapes de projection du RAG sont illustrées dans la figure 3.8. Considérons Res^l la résolution au l^{eme} niveau de la pyramide. Les blocs dans ce niveau sont de tailles variables $s \times s$ avec $s \in \{2^l, \ldots, 2^L\}$. Res^l est composée de blocs $B_{i,j}^{l,s}$ avec $s \times s$ la taille du bloc et (i, j) les coordonnées du bloc dans le niveau l.

Maintenant, nous allons décrire le processus de projection du RAG d'un niveau l à un niveau $l-1$ (algorithme 1). Tout d'abord, on associe à chaque bloc l'étiquette de sa région à travers la fonction $Label(.,.)$. Ensuite, on se réfère à la décomposition quadtree pour détecter l'ensemble des blocs à diviser. On obtient alors deux ensembles : les blocs dits *divisibles* et les blocs *fixes*. Les blocs fixes sont des blocs dont la taille est supérieure à la taille des blocs du niveau $l-1$, soit $s > 2^{l-1}$. Ces blocs correspondent aux feuilles dans l'arbre quadtree.

Pour notre part, le traitement se concentre sur les blocs divisibles. Différentes opérations vont être appliquées afin de mettre à jour le RAG en prenant en

Algorithm 1: Update RAG

```
input  : Res^l, RAG^{l+1}
output: RAG^l
foreach Reg_i in RAG^{l+1} do
    foreach Block_j in Reg_i do
        Label (Block_j, Reg_i)
        if isSplit (Res^l, Block_j) then
            remove Block_j from Reg_i
            foreach sub_block_k in Block_j do   /* getting from Res^l */
                newRegion = CreateNewRegion (sub_block_k)
                addRegion (newRegion, RAG^l)
            end
        end
    end
    if SizeFixe(Reg_i) > 0 then                 /* inherited regions */
        addRegion (Reg_i, RAG^l)
    end
end
UpdateNeghborhood (RAG^l)
Merge (RAG^l)
```

compte les nouvelles informations apportées par la pyramide. L'algorithme présente le processus de mise à jour du RAG lors de sa projection. Les blocs divisibles (vérifiant le prédicat $isSplit(.)$) sont décomposés en quatre sous blocs et leur information de couleur est récupérée depuis l'image Res^{l-1}. Ces blocs sont enlevés des régions auxquelles ils appartenaient et ils sont considérés comme de nouvelles régions (la fonction $CreateNewRegion(.)$) ajoutées au RAG.

Ensuite, les régions obtenues dans le RAG au niveau l sont projetées au niveau $l-1$ en fonction des blocs fixes qui les composent. En fait, chaque région qui contient au moins un bloc fixe (fonction $SizeFixe(.)$) est maintenue dans le RAG. Ainsi, ces régions sont projetées avec leur blocs fixes dans le RAG du niveau $l-1$. En plus, les régions projetées conservent leurs étiquettes, ce qui assure une certaine cohérence inter niveaux dans la pyramide. Les blocs divisés lors de la projection sont considérés comme des régions indépendantes, et seront soit fusionnées avec les régions fixes afin d'affiner leur contours, soit regroupées avec d'autres régions.

Cette projection de régions au niveau l fournit une répartition initiale des régions en tenant compte des segmentations précédentes. A cette étape, le

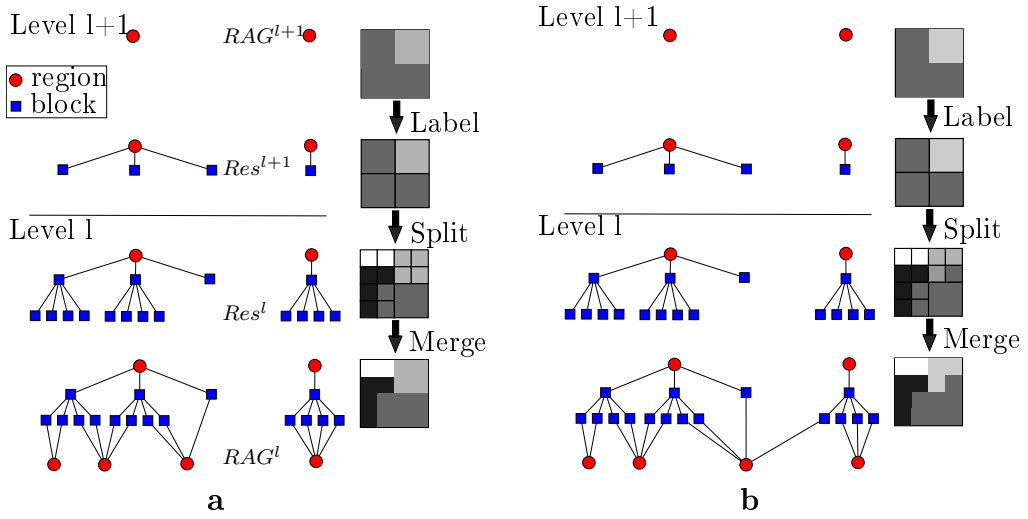


Figure 3.8 – Projection du RAG : relations inter-niveaux de quatre régions dans Res^l. (a) Les régions sont composées de blocs du même parent. (b) Une région est composée de blocs appartenant à des parents différents.

processus de segmentation va effectuer les mises à jour dans le RAG. Trois cas sont alors possibles :

1. Une région peut garder l'ensemble des blocs obtenus dans la segmentation du niveau précédent : les changements se feront autour de son contour en ajoutant éventuellement des blocs de petite taille pour le raffiner. Ces régions correspondent généralement à des grandes surfaces dans l'image à faible variation énergétique (exemple : murs, sol, portes...)
2. Une région peut être décomposée partiellement en gardant une partie de ses blocs fixes. Cette division est due à la perte d'information et de détail de contours dans les hauts niveaux de la pyramide. Cette perte est récupérée à mesure qu'on augmente la résolution de l'image.
3. Enfin, une région peut être complètement perdue, et voir tous ses blocs divisés. Ces régions sont principalement localisées dans des endroits texturés de l'image, là où il existe un amas de blocs de petites tailles 2×2 et 1×1 dans le quadtree.

Enfin, les régions du RAG sont fusionnées ($merge(RAG^l)$) en fonction de deux seuils T_{moy} et T_{grad}. T_{moy} est le seuil de fusion de la différence des moyennes entre deux régions et T_{grad} est le seuil de fusion du gradient entre les deux régions. Ce processus est itératif et permet de fusionner les régions en mettant à jour les liens du RAG jusqu'à ce que plus aucune fusion ne soit possible.

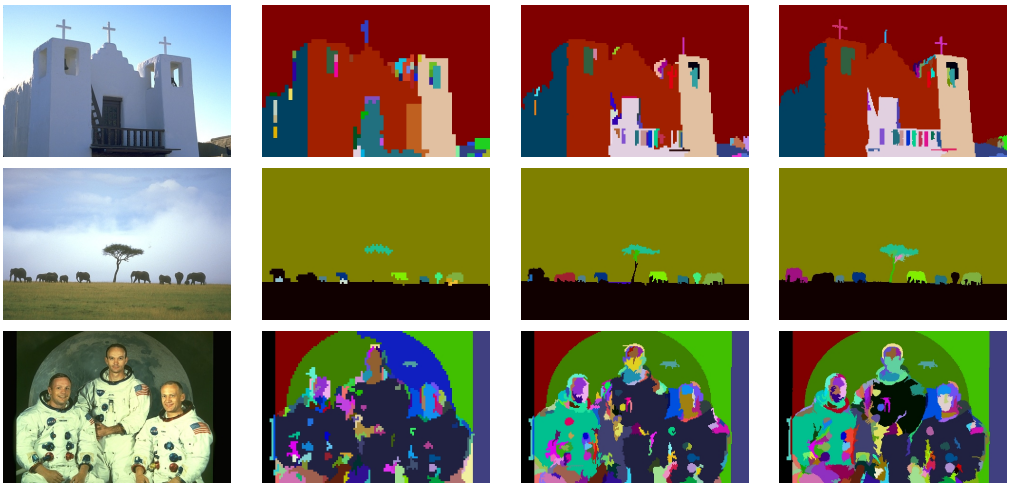


Figure 3.9 – Resultats de segmentation à différents niveaux de résolutions : 1×1, 2×2, 4×4.

La projection du RAG d'un niveau de résolution à un autre produit deux types de relation inter-niveaux entre les régions :

1. la création de nouvelles régions issues de la même région parente dans le niveau précédent (cf. figure 3.8(a)),

2. la création de nouvelles régions avec des blocs qui appartenaient à des régions différentes (cf. figure 3.8(b)).

La figure 3.9 présente les résultats intermédiaires à chaque niveau de résolution. Pour des raisons de visibilité, nous avons appliqué un sur-échantillonnage sur les niveaux de la pyramide. De gauche à droite, on peut observer les résultats de segmentation au niveau 3, 2, 1 de la pyramide régulière avec des seuils de fusion de la moyenne des régions à $T_{moy} = 24$ et le seuil du fusion du gradient entre deux régions $T_{grad} = 8$ et le seuil de découpage quadtree $T_y = 20$. On remarque que les contours sont affinés à mesure qu'on descend dans la pyramide. De plus, la cohérence des étiquettes d'un niveau à un autre est bien préservée grâce à l'algorithme de projection du RAG.

Maintenant, nous allons présenter la création de la pyramide de graphe qui permet de fournir des résultats à plusieurs niveaux de hiérarchie.

3.3.3 Segmentation hiérarchique

Le processus de segmentation JHMS effectue un regroupement hiérarchique à chaque niveau de résolution. Pour ce faire, les critères de similarité sont modifiés de telle sorte que la fusion soit de plus en plus tolérante entre les régions. Ceci permet de relâcher les contraintes de fusion et donc de réduire le nombre de régions et d'aller vers une représentation simplifiée de la carte de segmentation.

Le premier niveau de la hiérarchie correspond au graphe obtenu après projection du RAG du niveau précèdent. Ensuite, on applique un processus de fusion de régions similaires en fonction d'un critère d'homogénéité propre à chaque niveau de hiérarchie. Cependant, on ne peut pas se permettre de trop augmenter la valeur des seuils de fusion au risque d'autoriser des fusions non souhaitables puisque on ne contrôle pas l'ordre des fusions des régions. C'est pourquoi, si après la segmentation hiérarchique, on souhaite contrôler le nombre de régions, on va rechercher les deux régions connexes dans le RAG les plus similaires au lieu de se contraindre par un seuil. Ceci va définir l'ordre de la fusion des régions en donnant la priorité aux régions connexes qui minimisent leur distance au sens du critère d'homogénéité.

A chaque itération, on produit une nouvelle représentation en région avec un nombre réduit de régions. Cet empilement fourni la pyramide de graphe et donc une représentation hiérarchique de la segmentation. La figure 3.10 présente les résultats obtenus à partir de la segmentation hiérarchique en utilisant $T_y = 20$, $T_{moy} = \{6, 12, 18, 24\}$ et $T_{grad} = \{2, 4, 6, 8\}$. De gauche à droite, le nombre de régions est réduit et on remarque que les objets sémantiques sont de mieux en mieux représentés.

Dans la section suivante, nous allons présenter les critères de fusion utilisés lors de la segmentation.

Descripteurs et critères de fusion

Afin de décider la fusion de deux régions, deux critères de similarité sont utilisés : la couleur et le gradient. Nous travaillons à cet effet dans l'espace YUV qui offre une meilleure représentation de la scène. Le gradient est estimé sur les blocs appartenant aux frontières se trouvant entre deux régions.

Le choix de ces critères est réalisé en vue de répondre au critère de faible complexité. En effet, la couleur est obtenue directement par accès au contenu du pixel. Elle est mise à jour avec le gradient durant le processus de segmentation. Considérons R_i et R_j deux régions voisines dans le RAG. On définit la mesure

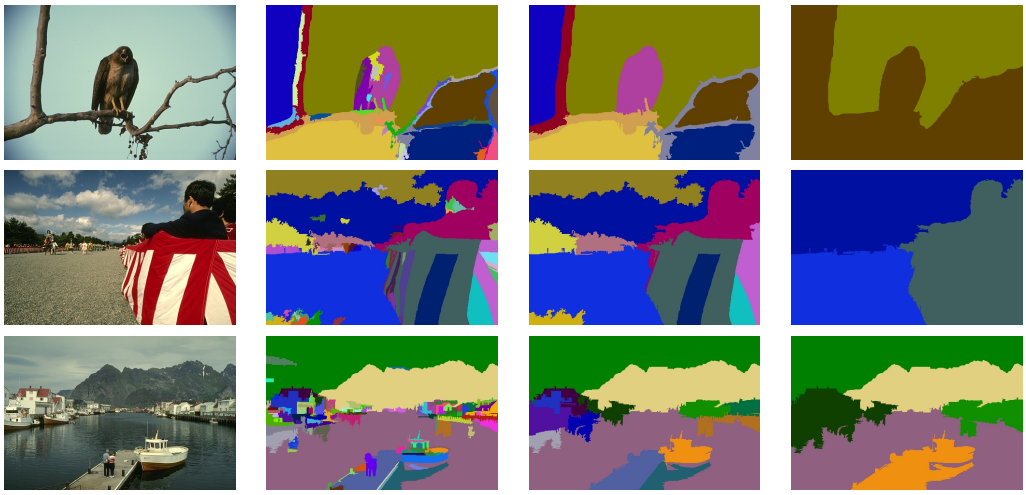


Figure 3.10 – Résultat de la segmentation hiérarchique

suivante :

$$\begin{aligned} Diff_{M_i,M_j} = \alpha \times (abs(M_i^Y - M_j^Y)) \\ + \beta \times (abs(M_i^U - M_j^U)) \\ + \gamma \times (abs(M_i^V - M_j^V)) \end{aligned} \tag{3.1}$$

avec M_i^Y, M_i^U et M_i^V les moyennes de luminance Y et chrominances U, V sur l'ensemble des blocs appartenant à la région R_i. Les coefficients α, β et γ sont fixés empiriquement afin de répartir les poids de contribution de chaque composante Y, U et V dans la mesure de similarité. De façon à obtenir une qualité visuelle suffisante, nous fixons conformément à [DeSimone 08] selon :

$$\alpha = 0.8 \qquad \beta = 0.1 \qquad \gamma = 0.1$$

De ces informations, nous pouvons extraire les contours des régions, on note $C_{i,j}$ le contour défini par un ensemble de blocs se trouvant entre les deux régions R_i et R_j. Soit $L(b)$ la fonction qui permet de retourner l'étiquette du pixel b. On note

$$C_i = \{b \in R_i | \exists\ b' \text{ connexe à b} : L(b) \neq L(b')\}. \tag{3.2}$$

Dans cette section nous avons présenté le schéma général de segmentation JHMS, ainsi que les outils et les descripteurs utilisés. Nous allons présenter dans la section suivante les résultats objectifs et visuels de notre technique.

3.4 Évaluation

Dans cette section nous allons présenter l'évaluation de notre méthode sur deux benchmarks de segmentation. Nous présentons aussi l'application de notre segmentation sur les images de type couloir afin de montrer son intérêt dans la tâche de représentation simplifiée de l'image couloir.

La segmentation proposée dans ce chapitre a été d'abord testée sur le benchmark de Berkeley [Martin 01]. Celui-ci est utilisé pour comparer les approches de segmentation basées généralement sur la détection des contours. Les contours de référence sont extraits manuellement, et dessinés à la main par un ensemble de personnes. Les cartes de segmentation manuelles sont superposées afin de construire l'image de segmentation vérité terrain de référence. La carte de segmentation de référence contient des contours avec des valeurs allant de 1 à 0 correspondant à leur cumul d'occurrences dans les segmentations lors de l'extraction manuelle.
Nous avons d'abord testé notre algorithme sur la base de données BSDS300, qui est composée de 300 images couleurs naturelles. La base de données est divisée en deux sous-ensembles : apprentissage et tests pour les algorithmes qui nécessitent une phase d'apprentissage pour fixer les bons paramètres. Enfin, le benchmark contient le résultat d'algorithme issus de l'état de l'art. Les résultats obtenus par le JHMS seront comparés avec ceux du benchmark.

Nous avons utilisé un deuxième benchmarck appelé SEISM [PontTuset 13]. Celui-ci introduit une nouvelle métrique qui permet d'évaluer la qualité des régions extraites par la segmentation. Il s'appuie sur des techniques de segmentation permettant de fournir des représentation pseudo-sémantiques de l'image. Les expérimentations sont effectuées sur la base de données BSDS500 présentée dans [Arbelaez 11] qui contient 500 images couleurs naturelles.

3.4.1 Résultats objectifs

Dans cette section nous allons présenter les expérimentations et les résultats de notre segmentation dans le deux benchmark cités précédemment.

Berkeley

L'algorithme de segmentation JHMS est basé uniquement sur la couleur et le gradient local. Afin d'effectuer une comparaison juste avec les algorithmes du benchmark, on compare notre segmentation avec des algorithmes qui utilisent les mêmes descripteurs.

Le meilleur score référencé est détenu par GPB [Maire 08] (Global Probability Boundary) : cet algorithme se base sur la détection des contours en estimant la probabilité a posteriori d'appartenance d'un point de contour à une région à fortes variations (texturée) ou à une variation locale entre deux régions différentes. Les contours fournis par le GPB ne sont pas forcément des contours fermés ce qui ajoute une étape supplémentaire si on veut extraire des régions.

Pour sa part, l'algorithme CG [Martin 04] (Color Gradient) utilise les mêmes descripteurs que le JHMS. Toutefois, les deux algorithmes précédents nécessitent des phases d'apprentissage afin de construire les modèles des contours des images. Les seuils de fusion du JHMS quant à eux sont fixés empiriquement suite à de nombreuses expérimentations sur des image tests issues de la base de données. Les valeurs retenues sont celles qui ont donné les meilleurs résultats visuels.

Dans le tableau 3.1, les scores objectifs sont affichés dans la base BSDS300 de Berkeley. Le meilleur score à atteindre est 0.79 qui correspond aux images de contours étiquetées vérité terrain. On remarque que le GPB fourni un score de 0.70 et le score de CG est 0.57. Le JHMS se place entre les deux en fournissant un score de 0.60 qui est meilleur que CG qui se base sur les mêmes descripteurs.

Algorithme	Vérité terrain	GPB	CG	JHMS
Scores moyens	0.79	0.70	0.57	0.60

Table 3.1 – Les scores objectifs du benchmark de berkeley de la database BSDS300

Le JHMS fournit une représentation pseudo-sémantique basée sur une approche de détection zone homogène dont les critères sélectionnés sont principalement la moyenne des couleurs et le gradient local. Le GPB a, quant à lui, été conçu pour détecter les contours dans l'image et donc plus adapté au benchmark qui consiste à évaluer directement les contours extraits. La différence du score entre GPB et JHMS est due au fait que GPB utilise des informations de textures sur des patchs afin de calculer la probabilité d'appartenance d'un point à un contour de région ou un point de haut gradient de la texture. D'autre part, le GPB offre une représentation hiérarchique des contours obtenus sur l'image pleine résolution.

Le score sur les contours qui est proposé dans [Martin 04] permet de repérer si il y a une sursegmentation de l'image ou au contraire si il y a une sous-segmentation. Cette technique purement basée contour contraint à considérer les régions par leur contour pour évaluer la qualité de la segmentation. Cependant, cette métrique ne permet pas de détecter des erreurs de contour incomplet. En

effet, si une petite partie du contour manque, alors son impact sur le score est faible. Ceci est principalement dû au fait que les techniques basées détection de contours extraient des contours qui ne sont pas toujours fermés. Or, dans une représentation régions, les contours délimitent les régions : un contour ouvert est synonyme de disparition de frontière entre deux régions et donc, au lieu de deux régions, on n'obtient qu'une seule région. D'où, la nécessité d'avoir une deuxième mesure complémentaire pour évaluer la qualité de la segmentation par rapport aux régions.

SEISM

Pour cela, nous avons comparé notre technique de segmentation dans le benchmark SEISM proposé récemment dans [PontTuset 13] qui permet d'évaluer les techniques de segmentation en fonction de leurs partitions et leurs contours extraits. Dans ce benchmark, on retrouve la même métrique de précision rappel sur les contours P_b et R_b proposée dans [Martin 04]. De plus, une nouvelle mesure précision rappel P_{op}, R_{op} a été proposée pour quantifier la précision et le rappel des régions obtenues par rapport à la vérité terrain. Ici, une région est interprétée comme étant soit un **objet** en fonction de son recouvrement avec une région vérité terrain, soit une **partie** dans le cas d'une sur-segmentation.

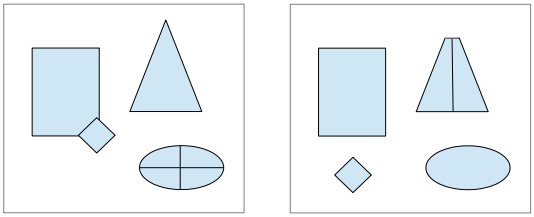
Considérons $S = \{S_1, S_2, \dots S_N\}$ la partition de l'image obtenue avec N régions et $G = \{R_1, R_2, \dots, R_M\}$ les régions vérité terrain. Pour estimer P_{op} et R_{op}, on calcule deux taux de recouvrement pour chaque paire de région S_i et R_j tels que

$$O_S^{ij} = \frac{S_i \cap R_j}{S_i} \qquad O_G^{ij} = \frac{S_i \cap R_j}{R_j}$$

Deux seuils sont alors définis : γ_o le seuil de recouvrement objet et γ_p le seuil de recouvrement partie avec $\gamma_o > \gamma_p$. En fonction de ces seuils, deux régions S_i et R_j sont classées selon O_G^{ij} et O_S^{ij} selon les règles suivantes :

- si $O_S^{ij} > \gamma_o$ et $O_G^{ij} > \gamma_o$, cela signifie que les deux régions S_i et R_j ont un fort recouvrement dans les deux directions. Alors les deux régions sont classées comme des **objets**,
- si $O_S^{ij} > \gamma_p$ et $O_G^{ij} > \gamma_o$ cela signifie que S_i se trouve globalement dans la région R_j (sursegmentation). Alors la S_i et considérée comme une **partie** et R_j un **fragment**,
- a l'inverse, si $O_S^{ij} > \gamma_o$ et $O_G^{ij} > \gamma_p$, ce qui signifie R_j est incluse dans S_i (sous-segmentation), on obtient R_j une **partie** et S_i un **fragment**.

La figure 3.11 présente une illustration de la classification du benchmark. A partir de cette classification, on calcul oc (resp. oc') le nombre d'objets de la



(a) segmentation S (b) vérité terrain G

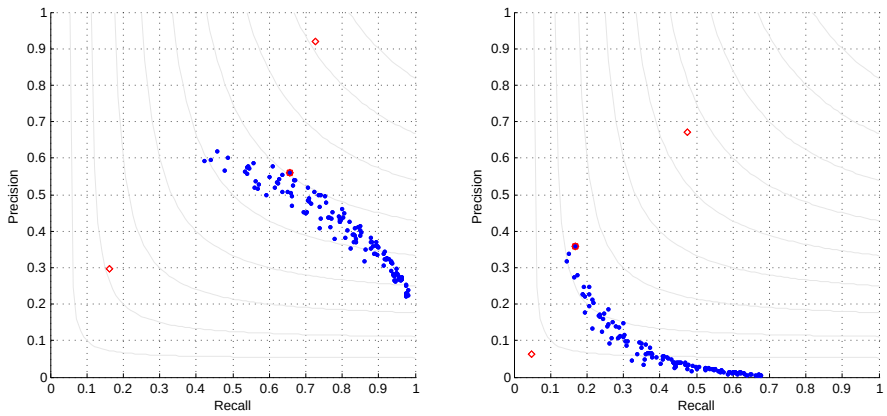
Figure 3.11 – Illustration de la classification des régions dans le benchmark SEISM [PontTuset 13] : les rectangles sont classés comme des objets grâce à leur recouvrement quasi total. Ensuite, l'ellipse est un fragment de G de taille 1 puisque il est recouvert de 4 parties de S. Le triangle est un fragment de S de taille 0.9.

segmentation (resp. de la vérité terrain) classés objets. fr (resp. fr') désigne la surface des fragments de la segmentation (resp. vérité terrain) couverts par les parties de la vérité terrain (resp. segmentation). Enfin pc (resp. pc') représente le nombre de parties de la segmentation (resp. de la vérité terrain). On peut estimer la précision P_{op} et le rappel R_{op} selon :

$$P_{op} = \frac{oc + fr + \beta pc}{|S|} \qquad R_{op} = \frac{oc' + fr' + \beta pc'}{|G|} \tag{3.3}$$

avec $0 < \beta < 1$.
On remarque que ce score pénalise une sursegmentation en multipliant le nombre de parties par β (plus le nombre de parties augmente, moins bon sera le score). Donc, dans un cas de sursegmentation on se retrouve avec un fort rappel et une faible précision. Par contre, dans le cas d'une segmentation grossière, on obtient un faible rappel et une forte précision.

Le JHMS est une technique de segmentation qui dépend de plusieurs paramètres : T_y le seuil de décomposition quadtree, T_{moy} le seuil de fusion moyenne, T_{grad} le seuil de fusion gradient et $nbHier$ le nombre de niveaux de hiérarchie. Nous avons d'abord chercher à trouver le jeu de paramètres le plus adéquat dans notre segmentation. Pour cela, nous avons lancé plusieurs expérimentations sur la base de donnée de test BSDS500 avec une recherche sur les différents paramètres cités avec des pas de quantification adapté afin de réduire le nombre de combinaisons possibles. Donc nous avons utilisé l'ensemble des paramètres suivants :



(a) contours (b) objets et partitions

FIGURE 3.12 – Résultats de la recherche du jeu de paramètre optimal : le point rouge dans le nuage de points correspond au meilleur score obtenu dans le benchmark

pour $T_y = 0$ jusqu'à 30 avec un pas de 5
  pour $T^1_{moy} = 2$ jusqu'à 10 avec un pas de 2
    pour $T^1_{grad} = 2$ jusqu'à 10 avec un pas de 2
      pour $nbHier = 1$ jusqu'à 5 avec un pas de 1
        JHMS$(T_y, T^1_{moy}, T^1_{grad}, nbHier)$

Ici, T^1_{moy} correspond au seuil de fusion du premier niveau de hiérarchie (de même pour T^1_{grad}). Les seuils au i^{eme} niveau de hiérarchie T^i_{moy} et T^i_{grad} sont obtenus selon :

$$T^i_{moy} = T^{i-1}_{moy} + T^1_{moy} \qquad T^i_{grad} = T^{i-1}_{grad} + T^1_{grad} \tag{3.4}$$

Par ailleurs, les paramètres du benchmark sont fixés par l'auteur à $\beta = 0.1$, $\gamma_o = 0.95$ et $\gamma_c = 0.25$

Les résultats sont ensuite évalués sur le benchmark SEISM. La figure 3.12 présente les précisions et rappels des scores sur les contours et les objets/parties. A partir de ces scores, on obtient la meilleure segmentation avec les paramètres suivant : $T_y = 20$, $T_{moy} = 6$, $T_{grad} = 2$, $nbHier = 5$. La sélection de la meilleure segmentation s'effectue en fonction de la F-mesure selon :

$$F_b = \frac{2(P_b.R_b)}{P_b + R_b} \qquad F_{op} = \frac{2(P_{op}.R_{op})}{P_{op} + R_{op}} \tag{3.5}$$

On sélectionne alors le jeu de paramètres qui maximise ce score.

D'une façon générale, on remarque que lorsque le seuil du gradient est inférieur à celui de la moyenne (approximativement sa moitié), on obtient des scores plus intéressants. De plus, il faut que le seuil T_{moy} dans le dernier niveau de hiérarchie soit supérieur au T_y : $(T_{moy}^{nbHier} > T_y)$ afin de pouvoir regrouper les blocs du quadtree avec un seuil plus tolérant que celui de la décomposition. Ainsi, on sera sûr de pouvoir fusionner des blocs pour éviter une sursegmentation de l'image.

Ensuite, nous avons comparé notre technique avec les algorithmes de segmentation référencés dans le benchmark. On trouve la technique *Ultrametric Contour Map* (UCM) [Arbelaez 11] qui est une extension du GPB. Elle consiste à retrouver les régions à partir d'une carte des contours en se basant sur une approche hiérarchique. *Efficient Graph Based Segmentation* (EGB) [Felzenszwalb 04] est une technique de segmentation basée graphe qui se base sur une processus de fusion entre deux régions seulement si la similarité et la dissimilarité entre ces deux régions sont acceptables. La technique du *Mean Shift* (Mshift) [Comaniciu 02] est une segmentation basée sur la classification des pixels en régions homogènes. La technique *Normalized Cut* (NCUT) [Shi 00a] consiste à retrouver la coupe minimale dans le graphe de régions. De plus, on retrouve deux autres techniques basées sur les arbres de partition binaires : *Normalized Weighted distance between Model with Contour complexity* (NWMC) [Vilaplana 08] et *Independent Identicallly Distributed -Kullback Leilbler* (IID-KL)[Calderero 10]

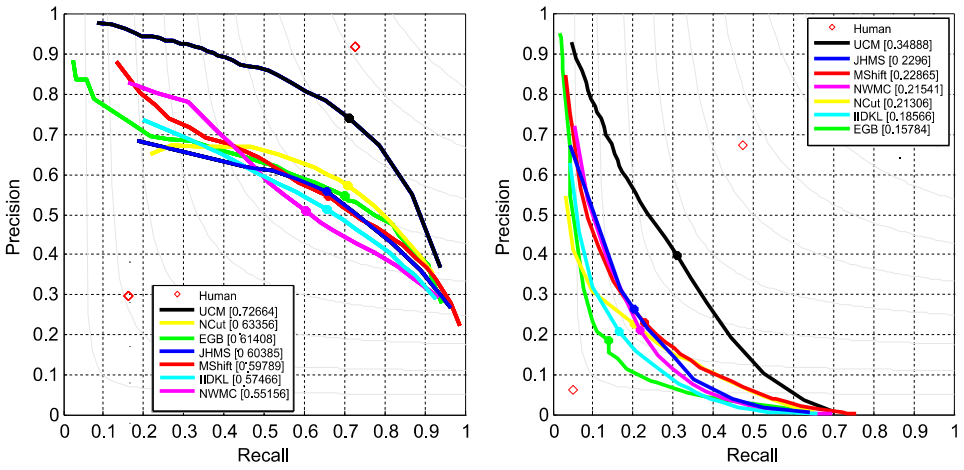
Afin de se comparer avec les autres techniques, nous avons utilisé le jeu de paramètres obtenu dans l'étape précédente, et nous avons fait varier le nombre de niveaux de hiérarchie afin d'aller d'une représentation sursegmentée vers une segmentation très grossière de l'image. Nous avons donc utilisé l'ensemble des paramètres suivant :

$T_y = 20$, $T_{moy} = 6$, $T_{grad} = 2$
pour $nbHier = 1$ jusqu'à 10 avec un pas de 1
    JHMS$(20, 6, 2, nbHier)$

Les détails des paramètres de générations des résultats des techniques référencées se trouvent sur le site internet du benchmark [1]. Les résultats obtenus sont représentés dans la figure 3.13 Les légendes des deux figures classent les techniques de segmentation selon la F-mesure. Ce score est estimé sur le meilleur résultat précision rappel.

La première chose observable est que le classement des techniques varie

1. `https://imatge.upc.edu/web/resources/supervised-evaluation-image-segmentation`



(a) contours (b) objets/parties

Figure 3.13 – Comparaison du JHMS avec les techniques de segmentation. Les scores associés à chaque technique dans les légendes représentent la $F-mesure$

entre les deux scores F_{op}, et F_b. La technique UCM est considérée comme la meilleure technique de segmentation dans les deux classements. Pour le reste, le classement d'une technique varie pour chaque score. Par exemple, le EGB, classé troisième technique en termes de précision/rappel des contours est rétrogradé en dernière position lorsque l'évaluation porte sur la qualité de ses régions comme objets/parties. Ceci est principalement dû au fait que l'algorithme fourni une représentation en contours qui ne sont pas toujours fermés, ce qui explique son bon résultat en F_b et pas sur F_{op}. En parallèle, notre technique JHMS est classée quatrième position selon F_b et en deuxième position juste après UCM en termes de qualité de segmentation F_{op}. Le classement des contours s'explique du fait que la représentation en régions obtenue est toujours en sur-segmentation par rapport à la vérité terrain. Ceci revient au choix des descripteurs utilisés dans notre technique (moyenne de couleur et gradient). En particulier, on ne prend pas en compte des descripteurs de texture lors de la fusion des régions. Le JHMS occupe la deuxième position en F_{op} du fait que cet algorithme de segmentation est basé sur la détection des régions homogènes : les régions obtenues sont délimitées par des contours fermés et sont toujours connexes. Ceci permet donc de s'assurer que les régions seront considérées comme des objets ou fragment suivant leur recouvrement avec la vérité terrain, et donc, un bon score de F_{op}.

Pour résumer, on peut dire que la segmentation JHMS est une technique très intéressante pour la segmentation des images naturelles. Les résultats objectifs

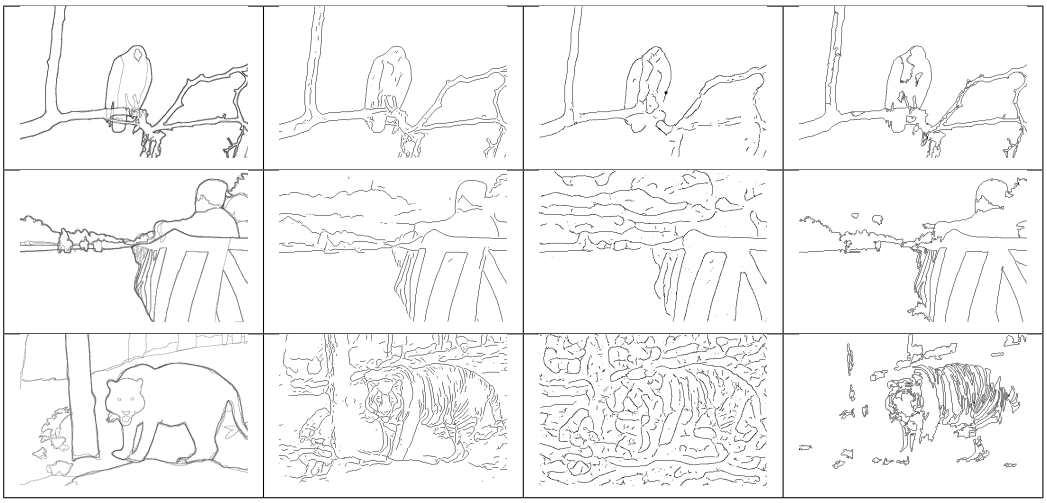


Figure 3.14 – Image de contours : de gauche à droite : vérité terrain, GPB, CG, JHMS.

montrent qu'elle peut surpasser ses concurrents en utilisant un ensemble de descripteurs faciles à extraire, tout en proposant la multirésolution et hiérarchie qui viennent enrichir la représentation en régions.

3.4.2 Résultats visuels

Les résultats visuels sont aussi intéressants et satisfaisants. La figure 3.14 montre des résultats de segmentation représentés par les images de contours issue de la base BSDS300. On peut remarquer que les contours détectés dans le JHMS (affichés dans la dernière colonne) sont quasi similaires aux contours vérité terrain. Cependant, les descripteurs JHMS restent classiques : ils n'utilisent pas de descripteurs globaux pour caractériser l'image et extraire notamment les textures. Ceci pénalise la segmentation en produisant une sur-segmentation sur, par exemple, l'image de tigre, qui dans GPB est considérée comme une région entière, alors que dans JHMS, chaque rayure sur sa peau est considérée comme une région à part.

Dans la figure 3.15, les contours détectés sont superposés directement aux images d'entrée afin de valider la cohérence des contours. L'algorithme utilise 3 niveaux hiérarchie.
Ici, $T_y = 20$, et les seuils de fusion utilisés sont :

$$T^1_{moy} = 8,\ T^2_{moy} = 16,\ T^3_{moy} = 24$$
$$T^1_{grad} = 4,\ T^2_{grad} = 8,\ T^3_{grad} = 12$$

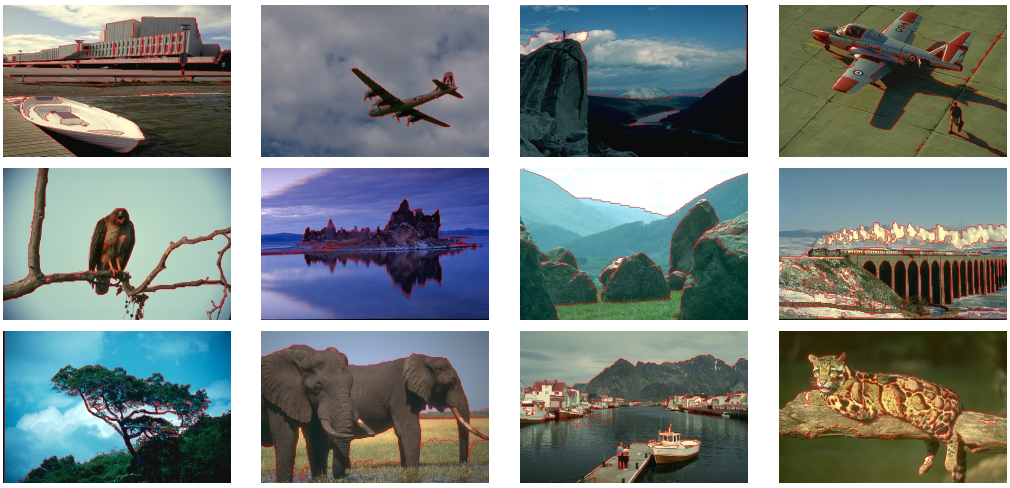


Figure 3.15 – Résultats de segmentation : cohérence des contours avec le contenu de l'image

3.4.3 Apport de la mulitrésolution à la qualité et la complexité de la segmentation

Afin de vérifier l'impact de la multirésolution dans la complexité en calcul, des tests ont été réalisés avec et sans multirésolution. La segmentation est effectuée dans une approche descendante, et est alors arrêtée à différents niveaux de résolution (avec des blocs, 1×x1, 2×2 etc).

Les tests sont effectués sur la base de données BSDS300 du benchmark de Berkeley. Les moyennes des temps d'exécution et les scores de segmentation sont rapportés dans le tableau 3.2. Trois différentes configurations de segmentation ont été réalisées :

- segmentation sans le quadtree (pleine résolution),
- segmentation multirésolution avec le bas de la pyramide qui correspond à la pleine résolution,
- segmentation multirésolution avec des niveaux demi/quart résolution.

Les résultats montrent que la multirésolution permet de réduire considérablement le temps de calcul tout en préservant la qualité. Cependant, raffiner la segmentation jusqu'à la pleine résolution (blocs 1×1) multiple le temps moyen de la segmentation par 5 : ceci se justifie au nombre important de blocs 1×1 qui peuvent se trouver dans le quadtree.

La multirésolution permet donc de réduire la complexité en calcul et aussi préserver la qualité visuelle et objective de la segmentation même si la segmentation

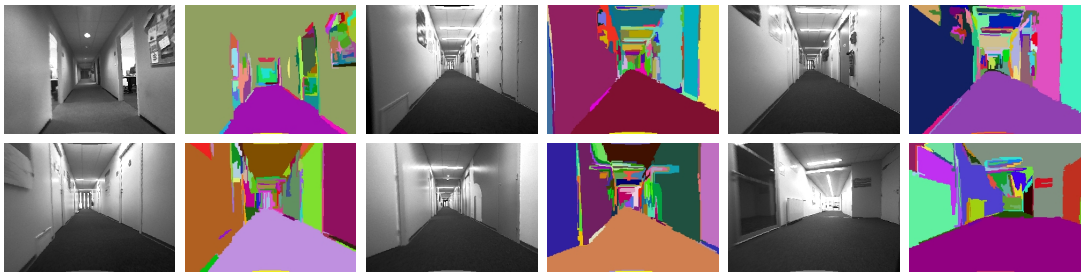


Figure 3.16 – Résultats de segmentation sur les images de couloirs.

est limitée en termes de quantité d'informations utilisées.

Scalabilité spatiale	Sans multirésolution	1x1	2x2	4x4
Scores (F_b)	0.57	0.586	0.589	0.589
Temps d'exécution (s)	0.445	1.517	0.311	0.108

Table 3.2 – Temps d'exécution moyens en secondes et scores de segmentation sur la base de Berkeley BSDS300

3.4.4 Segmentation des images de couloir

Nous avons aussi appliqué la segmentation JHMS sur des images de type couloir afin de vérifier que notre segmentation fonctionne aussi pour ce type de scène. Tout d'abord, les images obtenues à partir de la caméra embarquée sont monochromes (en niveaux de gris). Il y a peu de texture dans les images (sol, mur, portes...). Dans la figure 3.16, les résultats de plusieurs segmentations de différents couloirs sont présentés. Ces segmentations sont obtenues avec 5 niveaux de hiérarchie, les seuils de fusion du premier niveau de hiérarchies sont $T^1_{moy} = 2$ et $T^1_{grad} = 1$ avec un nombre de hiérarchie $nbHier = 5$.
Les seuils sont assez faibles parce que la scène est caractérisée par un faible gradient et un manque de texture. Les figures montrent que la technique est capable de bien détecter le sol dans toute les séquences (celui ci est de couleur différente et homogène). Les murs sont dans les images en blanc et souffrent de problèmes de réflexion de la lumière ; c'est pourquoi on détecte plusieurs régions sur les murs et le plafond. Grâce au faible seuil, les portes qui ont la même couleur que les murs peuvent être parfois bien détectées en régions à part.

Ces résultats montrent que pour avoir des segmentations intéressantes sur les images de couleur, il faut adapter le jeu de paramètres. Nous avons diminué les seuils afin de pouvoir détecter les régions tout en préservant les contours. La représentation obtenue est fidèle au contenu de l'image.

3.5 Conclusion

Dans cette section, nous avons présenté un nouvel algorithme de segmentation caractérisé par la multirésolution et la hiérarchie. L'algorithme est basé sur un partitionnement quadtree adapté au contenu. Afin d'exploiter au mieux la multirésolution, un algorithme de projection des RAG a été introduit afin de reconduire un résultat de segmentation d'un niveau de résolution à un autre. La projection du RAG offre plus de flexibilité dans la segmentation en autorisant les mises à jour de l'appartenance des blocs à une région à chaque fois qu'on ajoute de l'information.

Dans la suite de ce manuscrit, nous allons voir comment étendre cette technique de segmentation au cas des séquences vidéo.

Chapitre 4

Segmentation spatiotemporelle pour le suivi 2D

Dans le chapitre précédent, nous avons présenté un nouvel algorithme de segmentation d'images fixes dénommé JHMS. On a montré aussi que le JHMS est caractérisé par des propriétés de scalabilité tant au niveau spatial qu'au niveau sémantique. Dans ce chapitre, l'objectif est d'étendre cette segmentation d'images fixes vers la segmentation des séquences d'images.

La segmentation vidéo ou segmentation spatio-temporelle est l'opération d'extraction et de suivi de régions dans des séquences d'image. Les méthodes de segmentation vidéo ont toutes le même objectif : extraire des régions homogènes en fonction de certains critères prédéfinis.

Les techniques de segmentation tentent alors d'identifier les objets présents dans une scène en se basant sur les informations spatiales (luminance, couleur) et les informations temporelles (mouvement). Le résultat de la segmentation consiste alors en une représentation pseudo-sémantique composée de régions homogènes stables dans le temps. Cette représentation est définie de telle sorte qu'elle répond aux demandes et contraintes de son champ d'application. De ce fait, la notion d'objet vidéo reste purement subjective et dépend du contexte d'utilisation.

Dans le contexte de la navigation visuelle de robot mobile, la segmentation spatio-temporelle peut être utilisée dans plusieurs tâches : la détection des objets d'intérêt pour la localisation ou encore la détection d'obstacles dans la scène afin de déclencher des manœuvres d'évitement d'obstacles. Travailler dans le domaine de la navigation implique des contraintes de temps réel, et donc des complexités faibles en termes de calcul, surtout si les solutions sont destinées à être sur des

plates-formes robotiques où les puissances de calculs sont limitées.
La segmentation spatio-temporelle joue ainsi un rôle essentiel dans les méthodes d'analyse de l'information. En effet, elle est considérée comme un pré-traitement nécessaire pour fournir une représentation sémantiquement exploitable dans les domaines de la reconnaissance, de la classification et du suivi d'objet.

Ces travaux ont été réalisé au cours d'un séjour scientifique dans le laboratoire *Image Processing Group* à l'université polytechnique de Catalogne en collaboration avec Pr. Ferran Marques.

Ce chapitre se divise en trois sections principales. Tout d'abord, un état de l'art résume les principales approches de segmentation spatio-temporelle (4.1). Ensuite, nous présentons notre contribution dans la section 4.2, à savoir une segmentation spatio-temporelle basée sur la projection des contours. Enfin, les expériences et résultats obtenus sont présentés dans la section 4.3 afin de valider notre approche.

4.1 État de l'art

L'état de l'art nous propose un ensemble de techniques et d'approches de segmentation vidéo diverses et variées. Mergret regroupe les techniques de segmentation vidéo en trois classes selon leur stratégie de traitement [Megret 02].

- *Segmentation à priorité spatiale :* elle consiste à effectuer une segmentation en intra des images de la séquence, avant d'appliquer un suivi des régions afin de garantir la cohérence spatio-temporelle des régions.
- *Segmentation à priorité temporelle :* dans cette deuxième famille, l'approche effectue d'abord un premier regroupement des trajectoires des pixels dans la vidéo. Ensuite, pour chaque groupe de trajectoires, une segmentation spatiale est appliquée afin d'extraire les régions homogènes.
- *Segmentation jointe spatio-temporelle :* cette dernière famille représente un pixel par ses informations spatiales et temporelles. La segmentation consiste donc à détecter les régions temporelles stables dans la séquence d'images.

Nous proposons ici de classifier les techniques de segmentation en fonction de la manière de considérer la séquence d'images. On peut distinguer deux familles d'approches :

- *Segmentation 2D+T :* cette technique se base sur un traitement dans une fenêtre temporelle restreinte de la séquence d'images. Généralement cette fenêtre est comprise entre l'image courante I_t et l'image précédente I_{t-1}.

– *Segmentation 3D :* contrairement aux méthodes précédentes, les approches de segmentation 3D considèrent une séquence d'images comme un seul volume spatio-temporel de pixels, dans le but de traiter les dimensions spatiales et temporelles simultanément. Ainsi, la segmentation devient équivalente à la segmentation d'images 3D [Wirjadi 07].

On peut dès à présent se positionner dans cette classification. En effet, dans notre contexte, la séquence d'image est obtenue en temps réel ainsi que le traitement associé. On ne dispose donc pas de la séquence entière à un instant t. Ceci nous contraint donc à appliquer une segmentation de type 2D+T afin de suivre les régions dans les nouvelles images obtenues.

Dans la suite, nous allons présenter un ensemble non exhaustif de techniques de segmentations spatio-temporelle de type 2D+T et 3D. Ces techniques représentent généralement une extension des algorithmes de segmentation d'images fixes en ajoutant une dimension temporelle.

4.1.1 Segmentation 2D+T

Les approches dites 2D+T se focalisent d'abord sur l'extraction spatiale des régions. Ensuite, un suivi temporel est appliqué à travers une mise en correspondance des régions pour assurer la cohérence spatio-temporelle de ces régions.

L'idée globale de ce type d'approches est donc de projeter l'information de la partition obtenue à l'image I_{t-1} dans l'image I_t. Ce processus est effectué à travers la mise en correspondance des régions entre deux partitions qui peut se faire en fonction de critères de similarités spatiales et temporelles [Deng 98, DelBimbo 00, Deng 01], ou encore à travers une mise en correspondance de graphes d'adjacences de régions [Gomila 01, Galmar 05]. D'autres techniques se basent sur la co-segmentation qui consiste à segmenter le même objet à partir d'un ensemble d'images [Cheng 07, Rubio 13]. L'estimation de mouvement entre deux images est aussi utilisée pour projeter les régions d'une image à une autre [Marques 98, Park 00, Foret 02]. Cette partition projetée dans l'image I_t sera considérée comme une segmentation initiale de l'image I_t qui peut être par la suite affinée. Ceci permet au processus de segmentation d'assurer une continuité temporelle des étiquettes des régions.

Dans [DelBimbo 00], une segmentation vidéo est effectuée afin de fournir une représentation sémantique nécessaire à l'indexation de vidéos. Dans ces travaux, les auteurs effectuent d'abord une segmentation sur chaque image de la vidéo en utilisant la technique de clustering. Ensuite, un suivi de régions est effectué pour identifier les régions similaires. La mise en correspondance d'une région d'une

image I_t avec une région d'une image I_{t+1} est conditionnée par une mesure de similarité estimée en combinant à la fois la similarité en couleur (distance de Fisher entre les couleurs des deux régions exprimées dans l'espace L*u*v) et la similarité spatiale entre les deux régions (nombre de pixels en recouvrement). Cette technique simple n'offre un bon suivi de régions que si leur mouvement reste faible dans la séquence.

Une autre façon de faire correspondre deux cartes de segmentation consiste à utiliser le principe de recouvrement des germes [Deng 01]. Cette fois, les auteurs supposent qu'il existe un faible mouvement entre deux images successives afin d'assurer un large recouvrement entre les germes. Après extraction des germes de chaque image de la séquence, le suivi consiste à apparier les germes selon la procédure suivante :

1. Pour chaque germe, s'il existe un recouvrement avec un objet de l'image précédente, on lui assigne la même étiquette de l'objet. Sinon, on procède à la création d'une nouvelle étiquette et donc d'une nouvelle région.
2. Si un germe recouvre plusieurs objets de l'image précédente, les objets sont fusionnés.
3. Répéter 2 et 3 pour toutes les images de la séquence.

La mise en correspondance de graphes d'adjacences (RAG) a été présentée quant à elle dans les travaux de thèse [Gomila 01] en vue du suivi d'objets. La difficulté associée à la mise en correspondance des RAG est liée à la segmentation d'images qui est utilisée. En effet, il suffit d'un petit changement dans la disposition de la scène pour que les régions fusionnent dans un ordre différent. Ceci implique que deux RAG de deux images successives peuvent avoir un nombre de sommets et d'arêtes différents ce qui rend leur appariement difficile. Pour résoudre cela, Gomilia propose une phase de pré-traitement avant le processus d'appariement qui consiste à faire ressembler au mieux les deux RAG. Ce processus consiste à descendre dans le niveau de la hiérarchie des deux segmentations afin d'obtenir approximativement le même nombre de régions et la même disposition dans les deux partitions. Une autre technique de segmentation spatio-temporelle basée sur les RAG a été introduite dans [Galmar 05]. Ils imposent pour cela des contraintes de fusion en favorisant d'abord la fusion des régions à faible gradient dans une image I_t. Ensuite, un regroupement temporel est effectué entre les partitions de l'image I_{t-1} et celles de I_t sur les régions voisines spatio-temporellement pour affiner la segmentation et ainsi conserver la cohérence temporelle.

La projection des régions peut aussi se faire en estimant le mouvement de

chaque région. Dans [Park 00, Foret 02, Marques 98], la carte de segmentation de l'image I_t est projetée sur l'image I_{t+1} à travers une estimation de mouvement basée sur des algorithmes de type *block matching*. Ensuite, un ajustement de cette partition est effectué suivant le contenu de l'image : chaque région est projetée selon le mouvement estimé de ses blocs. Cette technique reste dépendante au choix de l'estimateur de mouvement utilisé qui peut engendrer des erreurs ou des ambiguïtés dans l'estimation de mouvement.

Les techniques basées sur la co-segmentation des partitions consistent à segmenter le même objet, la même scène à partir d'un ensemble d'images à travers des méthodes de classification [Rother 06, Glasner 11]. [Cheng 07] propose une application de la co-segmentation pour les séquences d'images. Ici la co-segmentation est effectuée entre deux images successives en adaptant une technique de classification semi-supervisée. Cette technique estime des mesures de similarité du même objet se trouvant dans différentes images. Ensuite, les région sont regroupées à travers des techniques de classification semi-supervisée.

Une dernière approche appelée *streaming hierarchical video segmentation* proposée dans [Xu 12b] considère une fenêtre temporelle un peu plus large. Pour cela, ils définissent des sous-séquences V_i de la vidéo V de longueur k images. Ensuite une segmentation hiérarchique basée graphe est effectuée pour chaque sous-séquence V_i de la vidéo. Cette segmentation produit une représentation $S_i = \{S_i^0, S_i^1, \dots S_i^h\}$ à h niveaux de granularité. Chaque niveau de la hiérarchie S_i^j de la sous-séquence V_i dépend de S_i^{j-1}, S_{i-1}^{j-1} et S_{i-1}^j à travers une hypothèse de Markov temporelle telle que :

$$S_i^j = argmin_{S_i^j} E(S_i^j | V_i, S_i^{j-1}, S_{i-1}^{j-1}, S_{i-1}^j, V_{i-1}) \tag{4.1}$$

avec $E(.|.)$ un modèle de segmentation conditionnel pour une sous-séquence. Un regroupement temporel est alors effectué pour attribuer les bonnes étiquettes des partitions S_{i-1}^{j-1} et S_i^{j-1} dans la partition S_i^j.

D'un point de vue global, le principal avantage des techniques 2D+T est le faible coût de calcul nécessaire pour le calcul de la segmentation. En effet, pour segmenter une image à l'instant t, seules les informations à l'instant t et $t-1$ sont nécessaires, ce qui implique une utilisation constante en taille de mémoire. De plus, puisque l'information de la segmentation est projetée d'une image à une autre, cela implique que ces techniques peuvent potentiellement traiter des vidéos de tailles importantes.
Cependant, puisque l'algorithme est séquentiel, il est primordial d'obtenir une bonne partition en régions pour appliquer le suivi, afin de minimiser l'erreur cumulée due à la projection des régions.

4.1.2 Segmentation 3D

Une approche 3D considère une séquence d'images en un seul volume pour réaliser l'extraction des régions volumétriques. Elle nécessite l'ensemble des images de la vidéo pour effectuer la segmentation. Parmi les approches de segmentation 3D (qui sont des extensions des techniques de segmentation d'images fixes), on trouve GB (*Graph Based*), une technique de segmentation d'image fixe présentée dans [Felzenszwalb 04] qui a été étendue pour former une solution à la segmentation vidéo GBH (*Graph-Based Hierarchical*) [Grundmann 10]. Une autre technique de segmentation d'images Meanshift [Paris 07] a été étendue à la segmentation vidéo Meanshift [Paris 08].

Dans [Greenspan 02], les auteurs proposent une approche statistique de regroupement de pixels de la vidéo par critère de similarité. Une classification non supervisée est alors effectuée dans l'espace des descripteurs spatiaux et temporels. Les pixels sont décrits dans un espace à six dimensions : trois composantes de couleur, deux dimensions d'espace et une dimension du temps. La modélisation est assurée par un modèle de mélanges de gaussiennes estimées à travers la technique de maximisation de l'espérance afin de permettre l'extraction des segments cohérents spatio-temporellement.

On retrouve aussi des techniques de segmentation basées graphe. Dans [Shi 98], la séquence d'images est représentée par un graphe pondéré reliant des patchs avec son voisinage spatio-temporel. En utilisant la technique des coupes normalisées (normalized cut), les partitions sont extraites en fonction des poids attribués aux arêtes qui correspondent à la somme des différence au carrés entre deux patchs voisins.

Afin de faire face à la complexité de calcul lors du calcul de la bonne coupe dans les graphes, [Fowlkes 01] propose une méthode basée sur une approximation utilisant la méthode de Nyström. Les auteurs exploitent le fait que le nombre de régions à extraire dans une séquence d'image est considérablement petit par rapport au nombre de pixels. Cette solution permet de réduire le problème en se basant sur un ensemble restreint de pixels de la séquence d'image.

Plus récemment, d'autres techniques exploitent la propriété de hiérarchie pour fournir une représentation riche à différents niveaux de sémantique. On retrouve l'approche GBH [Grundmann 10] qui utilise un algorithme basé graphe 3D hiérarchique. Le premier niveau de la hiérarchie contient uniquement des sommets, représentant des pixels et leurs relations de voisinage. Les relations d'adjacence sont spatio-temporelles et tiennent compte du flot optique : un pixel est adjacent aux pixels d'une image adjacente selon sa position estimée par le vecteur de mouvement. Une première passe de regroupement fournit une représentation initiale avec des petites régions spatio-temporelles. Cette représentation est utilisée pour construire une segmentation hiérarchique en se basant sur l'histogramme de

couleurs afin de sélectionner ensuite le niveau de granularité souhaité.

[Palou 13] propose une technique de représentation spatio-temporelle basée sur une regroupement de trajectoires représentées par des arbres binaires. Dans cette technique, les auteurs combinent à la fois les informations spatiales (distribution couleurs) et temporelles (trajectoire à long terme). Les trajectoires estimées à partir d'un estimateur de mouvement dense seront considérées comme une partition initiale de l'arbre binaire : tous les pixels à travers le temps appartenant à la même trajectoire forment une *région trajectoire*. Ensuite, les régions trajectoire sont fusionnées en fonction de leur similarité en couleur et en mouvement. Cette fusion va permettre de construire une hiérarchie des régions trajectoires représentée dans un arbre de partition binaire.

Les techniques présentées dans cette section fournissent des résultats intéressant et des partitions cohérentes dans le temps. Toutefois, leur principal inconvénient reste leur complexité de calcul à cause de la nécessité de traiter la vidéo en un seul volume. En effet, la technique de segmentation présentée dans [Palou 13] est d'une part en $O(NlogN)$ en termes de complexité de calcul avec N le nombre de régions. D'autre part, la technique est en $O(N)$ en termes de consommation de mémoire puisque la technique traite la séquence vidéo en un seul bloc. Par exemple, la technique est capable de segmenter une séquence contenant 3 million de voxels en 1000 secondes et nécessite 20GB de mémoire.

4.1.3 Discussion

Dans cette section, nous avons présenté un ensemble non exhaustif de techniques de segmentation vidéo. Le principal objectif de toute solution de segmentation est la cohérence spatio-temporelle des régions. Pour garantir cela, deux familles de techniques ont été présentées : 2D+T et 3D. La première consiste à traiter la séquence d'images séquentiellement en projetant la carte de segmentation d'une image à une autre. La deuxième famille traite la vidéo en un seul volume. On a montré que les techniques 3D sont très gourmandes en termes de consommation de mémoire et de calcul.

Dans notre contexte, il est donc exclu d'envisager une telle approche puisque nous ne disposons que de ressources limitées embarquées sur le fauteuil roulant. De plus, dans le cadre de la navigation, nous ne disposons pas de la séquence entière des images. C'est pour cela que nous avons opté pour une approche de type 2D+T.

Les techniques de 2D+T se basent donc sur la mise en correspondance de partitions entre deux images successives. Le choix de la stratégie de mise en

correspondance est le plus important dans une technique 2D+T pour assurer la cohérence spatio-temporelle des régions. Les techniques basées graphes souffrent de la non stabilité des graphes issus de deux segmentations différentes. De plus, la mise en correspondance des régions en fonction de leur recouvrement ne peut être effective que si le mouvement est faible entre deux images successives. Par ailleurs, les techniques basées sur l'estimation du mouvement instantané des pixels de chaque région sont sensible au choix de l'estimateur de mouvement utilisé. Enfin, estimer le mouvement dans des régions à faible gradient peut induire des ambiguïtés dans le mouvement estimé.

Nous avons ainsi opté pour une solution de suivi de régions basé sur le suivi de leurs contours. Nous avons imaginé une solution qui permet de détecter les zones de mouvements dans une image. Ces zones de mouvements vont représenter les zones sur lesquelles il faut appliquer la mise à jour des cartes de segmentation. En termes de complexité, cette solution ne nécessite le stockage en mémoire à un instant t que des données de deux images successives ainsi que les informations de leur partitions. Donc, la charge mémoire reste constante tout au long de la séquence ce qui est très intéressant dans contexte de navigation.

4.2 Contributions : Segmentation vidéo basée projection de contours

Dans cette section nous allons présenter notre contribution dans le domaine de la segmentation vidéo. Nous avons opté pour le choix de l'approche de segmentation 2D+T basée sur le mouvement (i.e segmentation + projection + mise à jour). Le schéma général de l'approche est présenté en section suivante.

4.2.1 Schéma général

Nous proposons ici une extension du JHMS présenté dans le chapitre 3. Afin de faire le suivi de région, nous allons essayer d'apporter une réponse à la question suivante : où les changements se produisent-ils ? Lors d'un mouvement dans une image, les régions vont subir une translation d'un point à un autre. Les régions étant délimitées par les contours, la translation des régions entraîne avec elle la translation des contours. Les changements dans une images sont majoritairement localisés dans le voisinage du contour. Par ailleurs, si le mouvement des régions est faible entre deux images, les régions vont se recouvrir en grande partie, et les changements vont s'effectuer au niveau des contours.

Considérons donc la projection des régions par la projection de leurs contours. La cohérence spatio-temporelle des régions sera assurée par le suivi de ces contours. La dimension temporelle est obtenue par une estimation du mouvement des contours entre deux images successives.

Le schéma général de la solution proposée est présenté dans la figure 4.1.

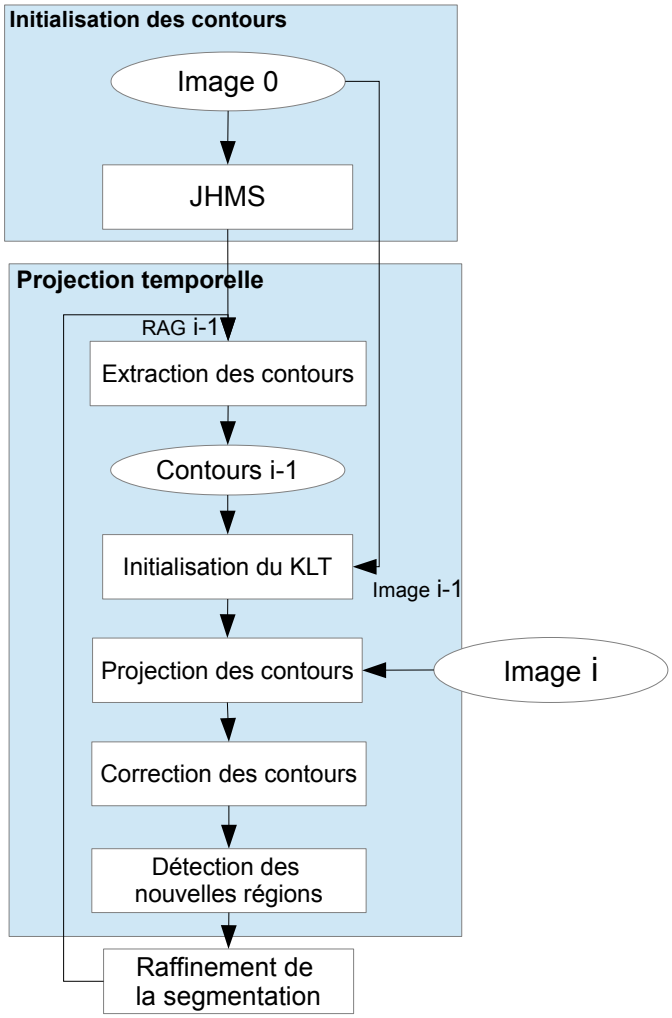


Figure 4.1 – Schéma général de la segmentation spatio-temporelle

1. Tout d'abord, une première segmentation est nécessaire pour l'initialisation des régions. On fait appel à la segmentation d'images fixes présentée dans le

chapitre 3. Les régions homogènes sont alors déduites de l'image en fonction de leur couleur et leur gradient local.

2. Cette première segmentation nous permet d'extraire les contours dans l'image. Ces contours vont servir à définir les régions et en même temps les suivre dans le temps. Chaque contour dans l'image est ensuite projeté dans l'image suivante en utilisant une approche KLT (section 2.2.1).
3. Le KLT est initialisé avec l'ensemble des points du contours. Ces points sont localisés dans des régions à fort gradient. Ceci permet un bon suivi avec un faible taux d'erreurs.
4. Ensuite, l'ensemble des contours est projeté dans l'image suivante et par conséquent les régions sont aussi projetées. L'utilisation de la projection de contours comme moyen de suivi de région permet de localiser les changements apparus dans la séquence.
5. Ensuite, une phase de modélisation et de correction du mouvement des contours est opérée afin de corriger les erreurs dues au KLT. Ces erreurs sont principalement dues au fait que le gradient observé autour des contours ne respecte pas forcement les propriétés des points à suivre par le KLT (forte variation bidirectionnelle du signal autour du point à suivre).
6. Afin de s'assurer que les nouvelles régions issues du mouvement de la caméra et des objets seront détectées, une étape de détection de nouvelles régions est effectuée. Cette technique repose sur le principe de différence des quadtrees entre deux images successives.
7. Un raffinement de la segmentation est effectué sur les régions issues de la projection des contours ainsi que les nouvelles régions. Une technique de croissance des régions projetées est appliquée en fonction des critères de couleur et de gradient afin de préserver la cohérence des étiquettes le long de la séquence.

Au niveau de la complexité, le suivi de région proposé se base sur la projection des seuls points de contours, ce qui s'avère nettement moins coûteux qu'un suivi dense de l'ensemble des éléments d'une régions.
Dans la suite, nous allons présenter en détails les différentes étapes de notre segmentation.

4.2.2 Initialisation

L'initialisation joue un rôle très important dans notre approche de segmentation spatio-temporelle. Une bonne extraction de régions avec des contours bien définis vont garantir un bon suivi de région le long de la séquence. Les contours

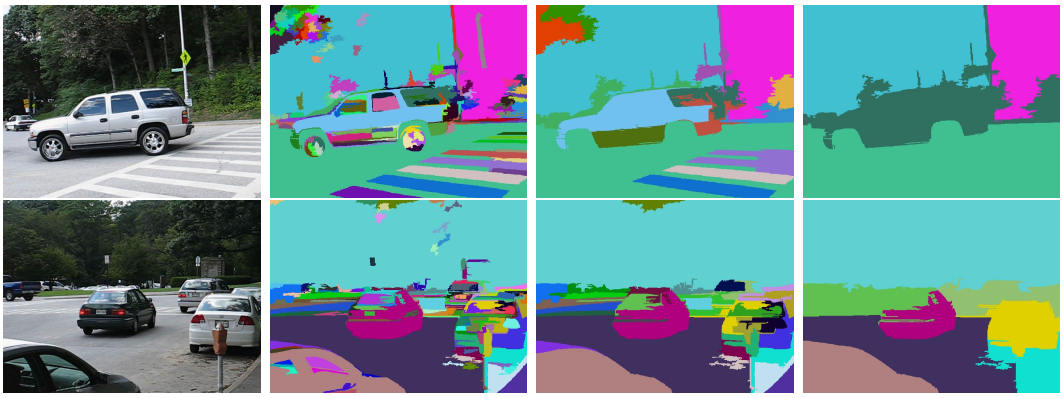


Figure 4.2 – Résultats du JHMS avec des représentations à différents niveaux de la hiérarchie

devront appartenir à des zones à fort gradient ce qui rendra leur suivi plus robuste. La segmentation requiert une partition initiale de la première image de la séquence. Cette première partition est effectuée avec le JHMS.

Afin de simplifier la représentation, le nombre de régions extraites est contrôlé lors de la construction de la hiérarchie. Ce contrôle s'effectue en fusionnant les régions voisines les plus ressemblantes en adaptant les seuils de fusion. La figure 4.2 présente les résultats de segmentation avec différents niveaux de hiérarchie (différents nombres de régions) avec $T_y = 20$, $T_{moy} = 6$, $T_{grad} = 2$ et $nbHier = 4$.

Extraction des contours

Les contours sont extraits à partir du RAG résultat de la segmentation de l'image initiale. Les contours sont les points connexes à deux régions. Selon la définition 4.2, les contours C_t d'une image I_t sont l'ensemble de pixels qui contiennent au moins un voisin qui appartient à une région différente. Ainsi,

$$C_t = \{p_x \in I_t | p_y \in N_{p_x} : L(p_y) \neq L(p_x)\} \tag{4.2}$$

avec N_{p_x} l'ensemble des pixels du voisinage 4-connexe autour du pixel p_x, et $L(p_x)$ la fonction d'étiquetage.

Pour rappel, l'algorithme JHMS possède plusieurs propriétés intéressantes :
- des contours caractérisés par un fort gradient parce que la fusion est basée sur le critère de gradient local ;
- des contours fermés puisque la segmentation d'image suit une approche par régions.

La deuxième propriété est très importante dans le suivi temporel de régions. Puisque nous allons définir les régions par leurs contours, il est donc indispensable que ces derniers soient fermés. Nous allons nous efforcer de préserver cette propriété tout au long de la séquence. Une fois que les contours sont définis dans une image I_t, l'objectif est de les reconstruire dans l'image suivante I_{t+1}. Pour ce faire, on procède en trois étapes : *projection, modélisation* et *correction.*

4.2.3 Projection des contours

La première étape consiste à projeter les contours d'une image I_t vers une image I_{t+1}. Chaque point de contour est projeté en utilisant l'algorithme KLT (1) présenté dans [Pressigout 06] qui fournit les vecteurs de déplacement $d = (dx, dy)$ pour chaque point du contour. Ainsi :

$$C_{t+1}(x, y) = C_t(x - dx, y - dy) \tag{4.3}$$

L'algorithme KLT peut toutefois échouer lors du calcul des vecteurs de déplacement des points de contours. On peut obtenir donc des erreurs de projection ainsi que des points non projetés. Pour pallier ce problème, il faut ajouter une étape de correction de contours.

4.2.4 Correction des contours

Même si les contours présentent de bonnes propriétés pour le suivi (du fait de la localisation sur des zones à fort gradient), ils sont cependant sujet à d'éventuelles erreurs de suivi ou de perte. Un bon point de suivi doit être, comme pour les coins de Harris, caractérisé par une forte variation bidirectionnelle. Or, les points utilisés pour notre suivi ne respectent pas cette condition.

Donc, afin de garantir une bonne reconstruction des contours, une étape de correction des points projetés est appliquée à l'ensemble des contours projetés. Ceci est réalisé par l'intermédiaire d'une étape de modélisation du déplacement des contours.

Estimation du modèle de déplacement du contour

Afin de récupérer les points de contours perdus, nous allons modéliser le déplacement de ces derniers en se basant sur les vecteurs de déplacement estimés lors du suivi KLT. A cet effet, on suppose qu'à une image I_t on dispose d'une bonne représentation en régions qui fournit des contours bien précis et fermés. L'objectif est d'assurer que ces contours vont suivre un mouvement cohérent lors

Algorithme 1 Algorithme KLT [Lucas 81b, Tomasi 91, Shi 94b].

Soient I_1, I_2 deux images rapprochées dans le temps, et W un patch à suivre centré au point $p_g = (x_g, y_g)$ suffisamment petit. On peut décrire alors le mouvement entre les deux images par une translation $t = (t_x, t_y)$. L'hypothèse de conservation de la luminance donne pour chaque point p_W du patch

$$I_2(p_W + t) = I_1(p_W) \tag{4.4}$$

Par voie de conséquence, le critère à minimiser est

$$C = \int \int_W (I_2(p + t) - I_1(p))^2 w(p) dp, \tag{4.5}$$

où $w(p)$ est une fonction de pondération appliquée au point p du patch. Dans le cas le plus simple, $w = 1$. Cela peut être aussi une fonction gaussienne centrée en p_g. Un développement de Taylor de premier ordre permet de linéariser et d'obtenir le système suivant :

$$Zt = a, \tag{4.6}$$

avec

$$Z = \int \int_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} w(p) dp, \tag{4.7}$$

où g_x est une gaussienne associée à la dérivée spatiale du signal lumineux en x. Le vecteur a est le vecteur d'erreurs observées sur la fenêtre de support, soit

$$a = \int \int_W (I_1(p) - I_2(p)) \begin{bmatrix} g_x \\ g_y \end{bmatrix} w(p) dp \tag{4.8}$$

La solution t est obtenue classiquement par une méthode de minimisation itérative de type Newton-Raspho.

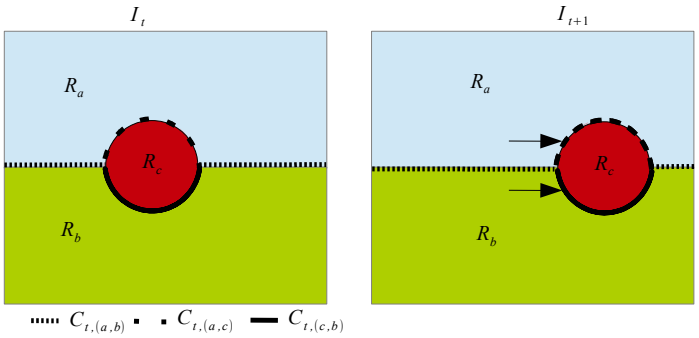


Figure 4.3 – Décomposition des contours des régions R_a, R_b et R_c en plusieurs parties $C_{t,(a,b)}$, $C_{t,(a,c)}$ et $C_{t,(b,c)}$.

de la projection tout en maintenant des contours précis et fermés.

Dans une séquence vidéo, on peut distinguer plusieurs types de mouvements (mouvement caméra et mouvement des objets). Pour modéliser les déplacements il est donc judicieux de décomposer les contours en sous-ensembles caractérisés par le même mouvement. En effet, les points des régions se trouvant en arrière plan doivent être regroupés. De même, pour chaque région appartenant à un objet en mouvement, un modèle est construit. Toutefois, à cette étape, il n'existe pas d'information sur les points de contours pour déterminer si le déplacement de ces derniers est dû au mouvement de la caméra ou au mouvement des objets.

Pour cela, nous proposons une décomposition des contours en se basant sur les différentes interactions qui existent entre les régions. Cette interaction varie pour chaque paire de régions (figure 4.3). Si on considère une région R_c comme un objet en mouvement. R_c est en voisinage avec plusieurs régions R_a et R_b. Dans cette courte séquence, on applique une translation de la région R_c vers la droite. Dans une configuration pareille, on peut distinguer deux modèles de mouvement : un premier modèle pour les points contours de R_c pour modéliser la translation, et un deuxième modèle pour les contours de R_a et R_b. N'ayant pas d'information sur l'objet en mouvement, on va décomposer les contours par des parties de contours se trouvant entre deux régions. Si on considère $C_{t,(a)}$, $C_{t,(b)}$ et $C_{t,(c)}$ les contours des régions R_a, R_b et R_c à l'instant t, nous allons construire alors trois modèles correspondant à chaque partie de contour se trouvant entre deux régions. $C_{t,(a,b)}$, $C_{t,(a,c)}$ et $C_{t,(b,c)}$ représentent les points de contours qui se trouvent entre les régions R_a, R_b et R_c, tels que :

$$C_{t,(i,j)} = C_{t,(i)} \cap C_{t,(j)} \quad avec \quad (i,j) \in \{a,b,c\}^2. \tag{4.9}$$

Les déplacements des contours sont ensuite modélisés par des mélanges

de gaussiennes afin de représenter toutes les composantes du déplacement qui peuvent exister dans un contour. Les paramètres du modèle sont estimés par l'algorithme d'espérance-maximisation [Bilmes 98] afin de déterminer les paramètres du maximum de vraisemblance d'un mélange de k gaussiennes.

On considère pour cela $\chi = \{\chi_0, \chi_1 \dots \chi_m\}$ l'ensemble des points de la partie contours $C_{t,(a,b)}$ à modéliser. Chaque point χ_i appartenant à $C_{t,(a,b)}$ est représenté dans $\mathbb{R}^4$ par ses coordonnées $(x,\ y)$ et son vecteur de déplacement $d = (dx, dy)$ tel que

$$\chi_i = (x, y, dx, dy).$$

La fonction de densité de la variable aléatoire χ est donnée par :

$$f(\chi_i|\theta) = \sum_{j=1}^{k} \alpha_j f_j(\chi_i|\theta_j) \tag{4.10}$$

$$\text{avec } f_j(\chi_i|\theta_j) = \frac{1}{\sqrt{(2\pi)^4\,|S_j|}} e^{-\frac{1}{2}(\chi_i-\mu_j)^T S_j^{-1}(\chi_i-\mu_j)} \tag{4.11}$$

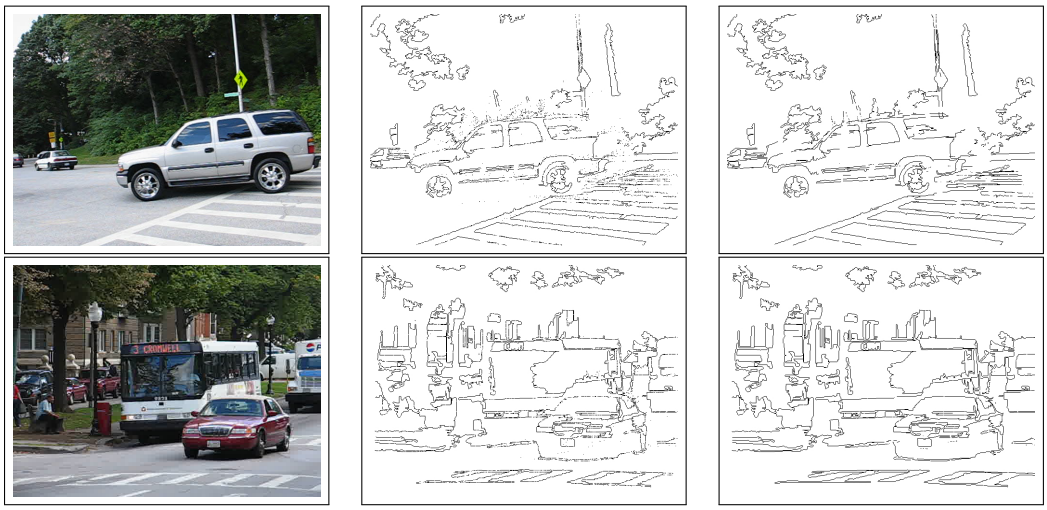
sachant que l'ensemble des paramètres $\theta = \{\theta_j\}_{j=1}^k = \{\alpha_j, \mu_j, S_j\}_{j=1}^k$ est défini comme suit :

- α_j est le poids de la j^{eme} composante gaussienne tel que $\alpha_j > 0$ et $\sum_{j=1}^{k} \alpha_j = 1$,
- $\mu_j(x_{\mu_j}, y_{\mu_j}, dx_{\mu_j}, dy_{\mu_j}) \in \mathbb{R}^4$ est la moyenne de la j^{eme} composante,
- S_j est la matrice de covariance de dimension 4×4 et de déterminant $|S_j|$.

Une fois l'estimation des paramètres effectuée, on peut appliquer la correction des déplacements des points de contours. Le déplacement $d_{\chi_i} = (dx_{\chi_i}, dy_{\chi_i})$ d'un point de contour χ_i est corrigé en fonction de moyenne de la composante gaussienne à laquelle il appartient, à savoir celle qui maximise la probabilité a posteriori $f_j(\chi_i|\theta_j)$ du point χ_i tel que :

$$d_{\chi_i} = (dx_{\mu_j}, dy_{\mu_j}) \quad \text{avec} \quad j = \text{argmax } f_j(\chi_i|\theta_j) \tag{4.12}$$

Dans la figure 4.4, on montre la différence entre les contours avant et après les corrections. En pratique, le nombre de gaussiennes est fixé à $k = 4$ et les poids $\alpha_j = \frac{1}{4}, j = \{1, \dots, 4\}$. La troisième colonne présente les zones de mouvement après la correction d'erreurs. On remarque que les contours sont précis et cohérents. Avec cette méthode, on peut assurer que deux points de contours voisins vont subir le même mouvement. De plus, les points perdus par le KLT sont récupérés en fonction du mouvement de leur voisinage. Ceci va assurer un mouvement global cohérent et préserver l'intégrité de la région.



(a) (b) (c)

Figure 4.4 – Correction des contours après projection. (a) image originale, (b) image des contours après projection KLT et (c) image des contours après correction des contours.

4.2.5 Détection des zones de mouvement

Les zones de mouvement jouent un rôle très important dans notre algorithme de segmentation. En effet, ces zones déterminent les endroits affectés par les mouvements dans la scène (cf. figure 4.5). Les zones de mouvement sont délimitées par les contours avant et après leur projection. Donc, si on considère $C_{t,(i,j)}$ le contour se trouvant entre les deux régions R_i et R_j de l'image I_t et $C_{t+1,(i,j)}$ le contour projeté et corrigé dans l'image suivante I_{t+1}, alors la zone de mouvement ZM est représentée par l'ensemble des pixels se trouvant entre les deux contours $C_{t,(i,j)}$ et $C_{t+1,(i,j)}$. On note

$$ZM = \{p_x \in I_{t+1} | p_x \in [C_{t,(i,j)}, C_{t+1,(i,j)}]\} \tag{4.13}$$

Dans la figure 4.6, on montre la détection des zones de mouvement à partir de deux images de contours successives. Les zones de mouvement sont déterminées soit par le mouvement de la caméra soit par le mouvement d'un objet. Dans le cas d'un mouvement de la caméra, les régions vont être caractérisées par un mouvement plus au moins similaire. Dans le cas des objets en mouvement, les zones de mouvement représentent des régions qui ont été occultées. L'objectif est de déterminer ensuite à quelles régions de l'arrière-plan les pixels de ces zones de mouvement vont appartenir. Ainsi, l'algorithme va procéder à des mises à jours bien localisées dans l'image, et va attribuer les pixels se trouvant dans les zones de mouvement aux régions obtenues dans la carte de segmentation précédente.

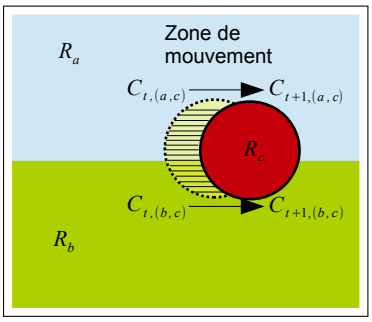
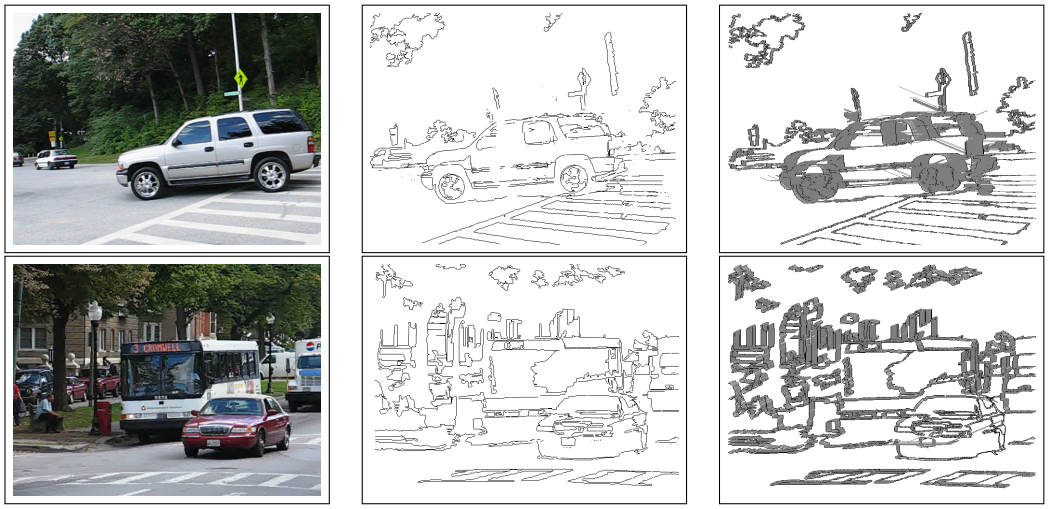


Figure 4.5 – Délimitation de la zone de mouvement : régions se trouvant entre les contours de départ et les contours projetés.



(a) (b) (c)

Figure 4.6 – Détection des zones de mouvement : (a) images originales, (b) Image des contours à l'instant et (c) les zones de mouvement associées (en gris).

Ceci permet de réduire considérablement la quantité de données utilisées lors de la segmentation, qui dépendra de l'ampleur du mouvement.

4.2.6 Détection des nouvelles régions

L'algorithme procède à une projection des contours de l'image afin de suivre les régions. Toutefois, il est impossible de détecter une nouvelle région dont les contours ne sont pas encore définis. Pour pouvoir détecter cette situation, nous avons fait appel à la représentation quadtree. En effet, on a vu que le quadtree permet de représenter les variations dans l'image dans le sens où plus la taille du bloc est importante, plus le bloc est homogène. Lorsque l'on a une zone homogène et qu'un contour apparaît (à cause du mouvement), alors le quadtree de cette

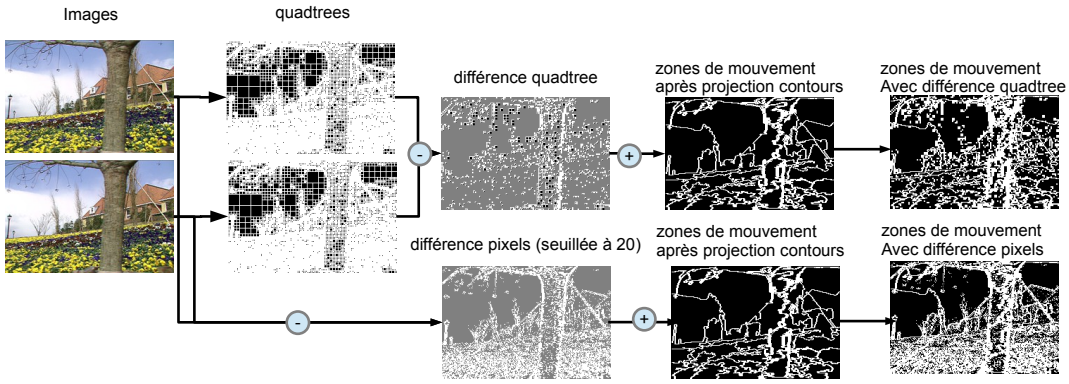


Figure 4.7 – Détection des nouvelles régions (comparaison entre l'application des la différence de quadtree et différence pixels)

zone qui au départ contenait des blocs de grandes tailles se décompose à cause du contour. Pour détecter les nouvelles régions, on applique donc une différence entre deux quadtrees de deux images successives (cf. figure 4.7). Cela corresponds à la détection des zones qui ont perdu leurs homogénéité lors de l'apparition d'un nouveau contour. Donc Si une nouvelle région apparaît, son contour provoque la décomposition des blocs du quadtree de la région homogène en plusieurs petits blocs. Ces nouveaux blocs vont tous être ajoutés aux zones de mouvement estimées auparavant. L'utilisation du quadtree est plus efficace que l'utilisation de la différence directe de deux images à cause des zones fortement texturées. En effet, un faible mouvement induit une forte variation dans les zones texturées lorsqu'on applique une opération de différence entre deux images successives (cf. figure 4.7). L'utilisation du critère de luminance pour la décomposition du quadtree offre une représentation indépendante de la couleur, les zones fortement texturée sont représentées par des blocs de taille minimale. Ainsi, même si un mouvement apparaît, les zones texturées conservent toujours des blocs de taille minimale et donc leur différence est nulle.

4.2.7 Raffinement de la segmentation

La projection des contours fournit les zones de mouvement. Ces dernières forment avec les nouvelles régions détectées et les régions de la segmentation de l'image précédente une première partition de l'image courante (figure 4.8.2). L'objectif dans cette dernière étape consiste à attribuer l'ensemble des pixels de la zone de mouvement aux régions déjà existantes. La décision d'attribution des pixels à telle ou telle région est prise en fonction des descripteurs de couleur du pixel et des régions avoisinantes. Différentes situations peuvent être gérées : création de nouvelles régions (figure 4.8.3), occultation partielle ou totale des

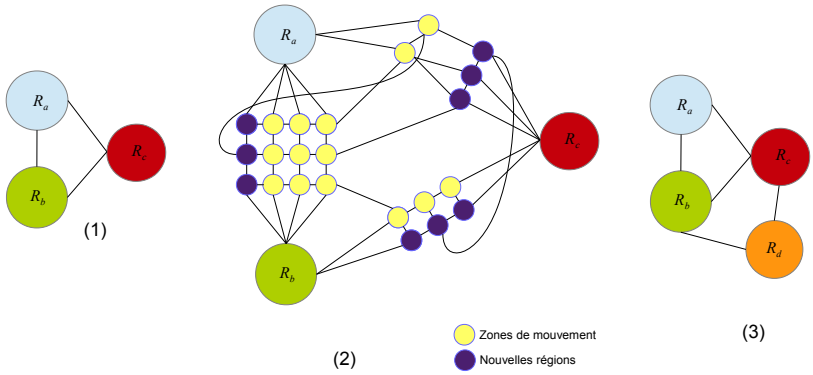


Figure 4.8 – Représentation RAG : (1) Régions de la segmentation à l'instant t. (2) Régions projetéed ainsi que les zones de mouvement et les nouvelles régions. (3) Nouveau RAG à l'instant $t+1$ avec apparition d'une nouvelle région.

régions. Le RAG reste donc flexible aux différents changements qui peuvent apparaître dans la scène.

La fusion se fait dans un ordre bien défini. Tout d'abord les régions projetées sont fusionnées avec les zones de mouvement et les nouvelles régions (cf. figure 4.9). Cette fusion se fait en fonction de la contrainte de voisinage et de ressemblance. Ensuite, les pixels restants (pas encore fusionnés avec les régions projetées) sont fusionnés entre eux pour former de nouvelles régions.
On constate donc que pour assurer la persistance d'une région, il faut un minimum de recouvrement entre deux images successives. Donc plus les régions sont grandes, plus le recouvrement est assuré. D'un autre côté, plus le mouvement est faible, plus les zones de mouvement sont petites (voire inexistantes) et donc plus le recouvrement est important. Ainsi, les étiquettes des régions sont projetées d'une image à une autre, la cohérence spatio-temporelle des régions est donc assurée.

Dans la suite de ce chapitre, nous allons présenter les résultats objectifs et visuels de notre technique de segmentation.

4.3 Résultats expérimentaux

Afin de valider la technique de segmentation spatio-temporelle présentée dans ce chapitre, nous avons comparé notre segmentation avec les résultats du benchmark présenté dans [Xu 12a]. Ce benchmark recense un grand nombre de techniques de segmentation spatio-temporelle (GB [Corso 08], GBH [Grundmann 10], SWA [Sharon 06], MeanShift [Paris 08], Nyström [Fowlkes 01]). L'avantage d'uti-

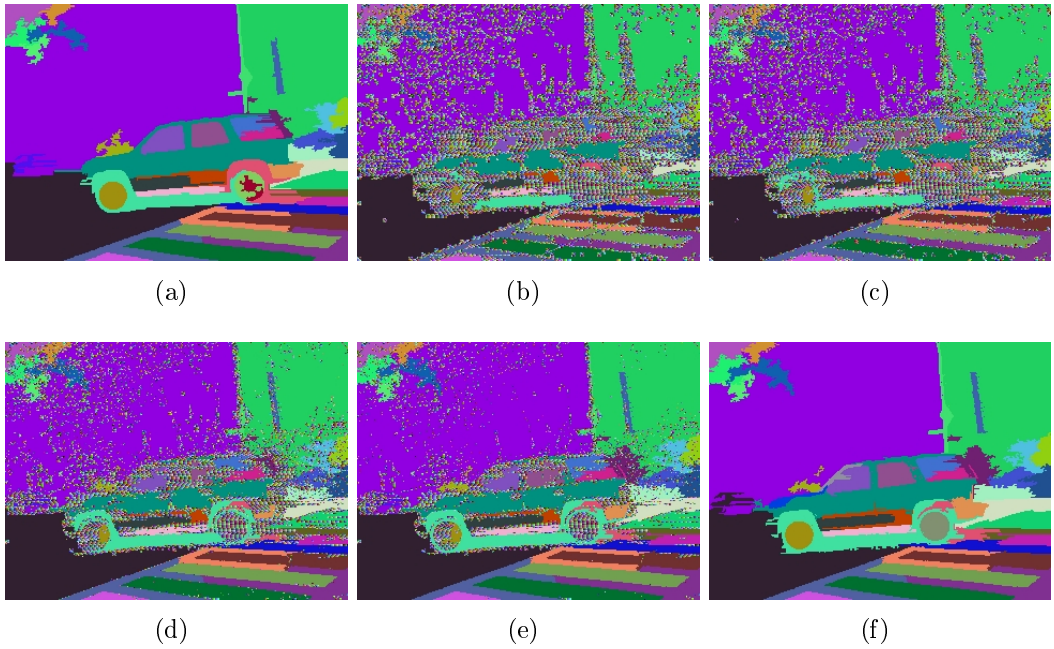


Figure 4.9 – Raffinement de la segmentation : fusion des régions projetées avec les zones de mouvement. (a) image segmentée à l'instant $t = 0$. (b) projection des régions à l'instant $t = 1$ ainsi que les zones de mouvement. (c-f) fusion progressive des zones de mouvement.

liser ce benchmark est que les techniques recensées sont très récentes et que le code du benchmark et ceux des techniques sont disponibles [1]. Ainsi, on peut positionner notre solution par rapport à l'état de l'art sur le plan de la qualité de la segmentation. Toutefois, les algorithmes référencés font partie uniquement de la famille des techniques 3D. Nous n'avons pas trouvé d'autre benchmark dédié aux techniques 2D+T. L'objectif ici n'étant pas de proposer un nouveau benchmark pour les techniques 2D+T, nous allons tout de même évaluer notre technique en utilisant les métriques du benchmark cité précédemment pour se comparer avec les techniques recensées.

En ce qui concerne les solutions 3D, on a vu dans l'état de l'art que ces techniques considèrent la vidéo en un seul volume 3D : ceci leur permet de créer des régions connaissant leur évolution dans la séquence. Cependant, notre technique est de type 2D+T, et donc traite la vidéo séquentiellement : segmentation puis projection puis raffinement. On a vu que ces techniques ne disposent que d'informations locales afin de repérer les changements entre deux images successives $(I_t,\ I_{t-1})$. Cela nous amène à dire que les techniques 3D ont un avantage non

1. `http://www.cse.buffalo.edu/~jcorso/r/supervoxels/index.html`

négligeable par rapport à notre technique de segmentation et donc une comparaison entre techniques 3D et 2D+T sera naturellement en faveur des techniques 3D.

Le benchmark utilise un ensemble de vidéos [Chen 10] qui représente une partie de la base de données Xiph[2]. Ces vidéos ont été annotées en 24 classes sémantiques. Ces classes sont les mêmes que celles de la base de données de la segmentation d'objets MSRC [Shotton 09]. La base de données est composée de 8 vidéos de 80 images en moyenne pour une résolution de 240 $\times$160.

Dans la suite, nous allons présenter les résultats objectifs, avant de mettre en lumière les résultats visuels de notre segmentation en les comparant avec les techniques existantes.

4.3.1 Résultats objectifs

Afin de pouvoir évaluer les résultats des techniques de segmentation, plusieurs métriques ont été définies afin d'évaluer la précision des régions et les contours obtenus. Une segmentation spatio-temporelle produit un ensemble de régions volumétriques appelées *supervoxels*. Les supervoxels sont des segments délimités dans l'espace par des contours et dans le temps par une durée de vie.
Maintenant, considérons une segmentation vérité terrain composée de m régions volumétriques $g_1, g_2, \ldots, g_m$ et une segmentation vidéo composée de n supervoxels $s_1, s_2, \ldots, s_n$. Plusieurs métriques sont alors introduites :

- **Erreur de sous-segmentation 3D (UE)** : cette métrique permet de mesurer la portion du supervoxel s_j excédant le volume de la région volumétrique vérité terrain g_i correspondante. La métrique s'obtient par

$$UE(g_i) = \frac{(\sum_{s_j \cap g_i \neq \emptyset} Vol(s_j)) - Vol(g_i)}{Vol(g_i)} \tag{4.14}$$

  où $Vol(g_i)$ est la fonction qui retourne le volume de g_i. Le score UE de la séquence est la moyenne des scores $UE(g_i)$ pour chaque volume vérité terrain.

- **Rappel sur les contours 3D (ACCU)** : cette métrique mesure le rappel des contours spatio-temporels (le taux des contours vérité terrain correctement détectés). Pour chaque couple g_i et s_j, les contours intra image et inter images sont superposés pour former un contour 3D.

2. `http://media.xiph.org/video/derf/`

– **Précision de la segmentation** : cette métrique permet d'estimer quelle portion du segment vérité terrain a été correctement segmentée par la technique. Ce score est estimé en fonction du volume de la région vérité terrain g_i couvert par des supervoxels qui ne couvrent pas d'autres régions. Pour ce faire, on défini $\overline{s_j}$ un supervoxel dont la majeure partie couvre g_i. On a :

$$ACCU(g_i) = \frac{\sum_{j=1}^{k} Vol(\overline{s_j} \cap g_i)}{Vol(g_i)} \tag{4.15}$$

La précision globale est estimée en moyennant les précisions de chaque volume vérité terrain $ACCU(g_i)$.

– **Durée moyenne** : cette métrique elle représente la durée de vie moyenne des supervoxels. Cette métrique permet de déduire la stabilité de la segmentation dans le temps, dans la mesure où les régions vont apparaître sur plusieurs images assurant la cohérence spatio-temporelle.

Dans un premier temps, nous avons testé la sensibilité de la technique de segmentation vis à vis des paramètres de fusion T_{moy} et T_{grad}. Nous avons fixé pour cela le nombre de hiérarchie $nbHier = 1$. Pour chaque séquence, nous avons fait varier la valeur T_{moy} et T_{grad} de façon incrémentale. Nous avons établi une relation entre ces deux valeurs suite à nos expérimentations dans le chapitre précèdent, soit :

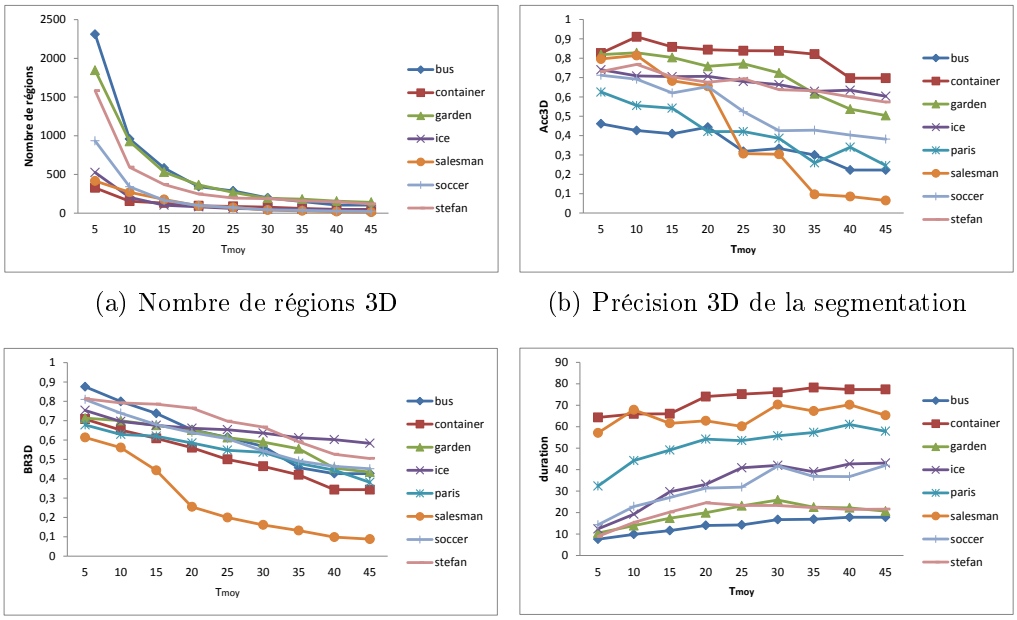
$$T_{grad} = T_{moy}/2. \tag{4.16}$$

T_{moy} prend des valeurs comprises entre 5 et 45 afin de passer d'une sursegmentation de la vidéo à une segmentation plus grossière. Nous avons généré 9 séquences avec un pas quantification du T_{moy} égal à 5 pour réduire le nombre de tests. Ensuite, nous avons relevé les différents scores spécifiques pour chaque type de séquence (*bus*, *container* ...).

Les résultats obtenus sont affichés dans la figure 4.10 : chaque figure correspond aux résultats associés à une métrique donnée.

Tout d'abord, la première figure présente la variation du nombre de régions en fonction du seuil de fusion. On peut remarquer que pour la même valeur du seuil, le nombre de régions varie selon le contenu de l'image (cf. figure4.10(a)). Par exemple pour $T_{moy} = 10$, on obtient 958 régions dans la séquences *bus* contre 154 régions dans la séquence *container*. En effet, plus la séquence est texturée, plus le nombre de régions augmente à des seuils bas.

La précision de la segmentation (figure 4.10(b)) représente le pourcentage de régions correctement segmentées relativement à la vérité terrain. Ici, on



(a) Nombre de régions 3D

(b) Précision 3D de la segmentation

(c) Rappel 3D des contours

(d) Durée des régions

Figure 4.10 – Résultats objectifs par séquence d'images

remarque que chaque séquence réagit différemment aux changements de la valeur des seuils selon son contenu. Par exemple, la séquence *salesman* est très sensible aux seuils de fusion et subit une forte chute dans le score de 0.8 à 0.1. Ceci est principalement dû au fait que cette séquence est caractérisée par un faible contraste et un mouvement très faible. Ceci implique que les régions correspondant à l'avant-plan vont être confondues avec les régions de l'arrière-plan. Par contre, pour des séquences comme *container*, on remarque qu'elles ne sont pas affectées par ce changement à cause de leur contenu caractérisé par des régions bien distinctes. Pour des séquences plus texturées (*bus*, *garden*), la précision diminue progressivement à chaque fois que le seuil est augmenté, les seuils de fusion deviennent de plus en plus tolérants.

Le troisième score est le rappel des contours : ce score représente le pourcentage de contours vérité terrain qui ont été détectés par la segmentation. Les scores pour chaque séquence sont représentés dans la figure 4.10(c). Les scores ici suivent à peu près la même tendance pour l'ensemble des séquences. Dans les seuils bas, on obtient des scores très importants, jusqu'à 0.9 de rappel des contours. Ceci s'explique par la sur-segmentation des séquences : en effet, les contours des régions vérité terrain sont quasiment tous détectés. Cependant, lorsque le seuil de

fusion est important, le score diminue à cause de la perte des contours : ceci est dû à l'augmentation de la valeur du seuil du gradient qui va autoriser à fusionner des régions qui présentent des gradients plus importants. Donc, plus les seuils de fusion sont petits, plus on parvient à détecter les contours des régions vérité terrain.

En dernier, la durée moyenne des régions est présenté dans la figure 4.10(d). Cette figure représente la cohérence temporelle des régions. Cette métrique dépend du mouvement qui caractérise la séquence : un mouvement important de la caméra implique l'apparition et la disparition des régions et donc une durée de régions faible. Par exemple, la séquence *container* contient au total 86 images : la scène est filmée par une caméra quasi fixe et présente un paysage ciel+mer avec un bateau qui navigue à faible vitesse impliquant peu de mouvement dans la séquence. La durée de vie des régions pour cette séquence est autour de 80 images. Cependant, dans la séquence *bus*, un mouvement de la caméra combiné avec un mouvement du bus provoque l'apparition et la disparition de nombreuses régions. Cette séquence contient 85 images et la durée des régions varie entre 20 à 10 images.

4.3.2 Comparaison avec les autres techniques

Nous avons montré dans le début de cette section qu'une comparaison d'une technique 2D+T avec des techniques 3D sera naturellement en faveur des techniques 3D.
Dans ce benchmark, les résultats des techniques référencées 3D ont été générés en contrôlant le nombre de régions. Ils supposent que les meilleures performances sont obtenues pour un nombre de régions compris entre 200 et 900 par séquence. Ce contrôle est rendu possible grâce à la hiérarchie : en traitant la vidéo entière, une technique 3D peut définir le nombre de régions pour l'ensemble des images de la vidéo à chaque niveau de hiérarchie.
Dans notre cas, il n'est pas évident de connaître dès le départ le nombre de régions que l'on va obtenir dans la vidéo. Même si pour la première image, le nombre de régions peut être fixé, ce nombre varie au long de la séquence du fait de l'apparition/disparition de régions. On obtient donc à la fin de séquence un nombre de régions différent de celui de départ, puisque nous fixons a priori le nombre de niveaux de hiérarchie.

Afin de se comparer avec les techniques de l'état de l'art, nous avons mis au point un dispositif de contrôle de la taille de régions afin d'avoir un nombre de régions comparables avec les autres techniques. Pour ce faire, nous avons fixé les seuils de fusion $T_{moy} = 20$, $T_{grad} = 10$ et le nombre de niveaux de hiérarchie $nbHiera = 1$. Ensuite, nous avons introduit deux nouveaux paramètres qui

correspondent aux tailles des régions $Size_{max}$ et $Size_{min}$. Le premier paramètre $Size_{max}$ permet de stopper la croissance des régions homogènes dès lors que la surface de ces régions atteignent la valeur $Size_{max}$: lors de la segmentation, les zones homogènes dans l'image grandissent en effet à une vitesse plus rapide que les autres (grâce à la multirésolution qui permet de fusionner les blocs de grandes tailles en premier). Le paramètre $Size_{max}$ permet donc de réguler la vitesse de croissance des régions. Le deuxième paramètre $Size_{min}$ permet d'éliminer les petites régions. Les tailles sont exprimées en nombre de pixels que contient une région dans une image t donnée.

Les résultats obtenus offrent des séquences d'images segmentées en un nombre de régions allant de 10 jusqu'à 4000 supervoxels selon les paramètres $Size_{max}$ et $Size_{min}$. Ceci rend alors les cartes obtenues comparables avec les techniques recensées dans le benchmark. Les scores sont affichés dans la figure 4.11. On peut remarquer que même si les techniques 3D sont plus avantagées que notre technique 2D+T, nous obtenons des scores comparables.

En termes de précision de la segmentation, 4.11(a), notre technique se classe derrière GB, GBH et MeanShift. Plus de 65% du volume est correctement segmenté. La baisse du score s'explique par la présence de la technique d'élimination des petites régions. Ainsi, les régions dont leur volume est inférieur au seuil $Size_{min}$ sont fusionnées avec leurs voisins le plus similaire en termes de moyenne et de gradient. Ces régions se trouvent généralement sur les contours, et peuvent être fusionnées avec le mauvais voisin, ce qui induit une chute des performances.

L'erreur de sous-segmentation est aussi faible (cf. figure 4.11(c)), et est égale au techniques SWA et GBH. Ceci s'explique par la technique de raffinement de la segmentation. En effet, l'utilisation de la technique de croissance de régions pour fusionner les régions permet de fournir des supervoxels qui respectent le contenu de l'image et ainsi débordent peu des régions vérité terrain.

La précision de contour (cf. figure 4.11(b)) est presque la même que le SWA et se situe loin devant les méthodes GB et MeanShift. Ceci est dû à la sur-segmentation qui préserve les contours des régions vérité terrain. De plus, l'utilisation du KLT pour projeter les contours est efficace : on remarque en effet que les contours sont conservés d'une image à une autre.

Cependant, la durée moyenne des régions (cf.figure 4.11(d)) est assez faible comparée aux techniques 3D. Ceci est principalement dû à la contrainte de taille de régions qui est très petite. En effet, projeter les contours puis appliquer un raffinement de la segmentation implique qu'il existe un minimum de recouvrement entre les régions. Cependant, lorsqu'on a des régions de petite taille, le recouvrement est faible, voire inexistant. C'est pourquoi nous observons une

faible durée moyenne de supervoxels à chaque fois que le nombre de régions augmente.

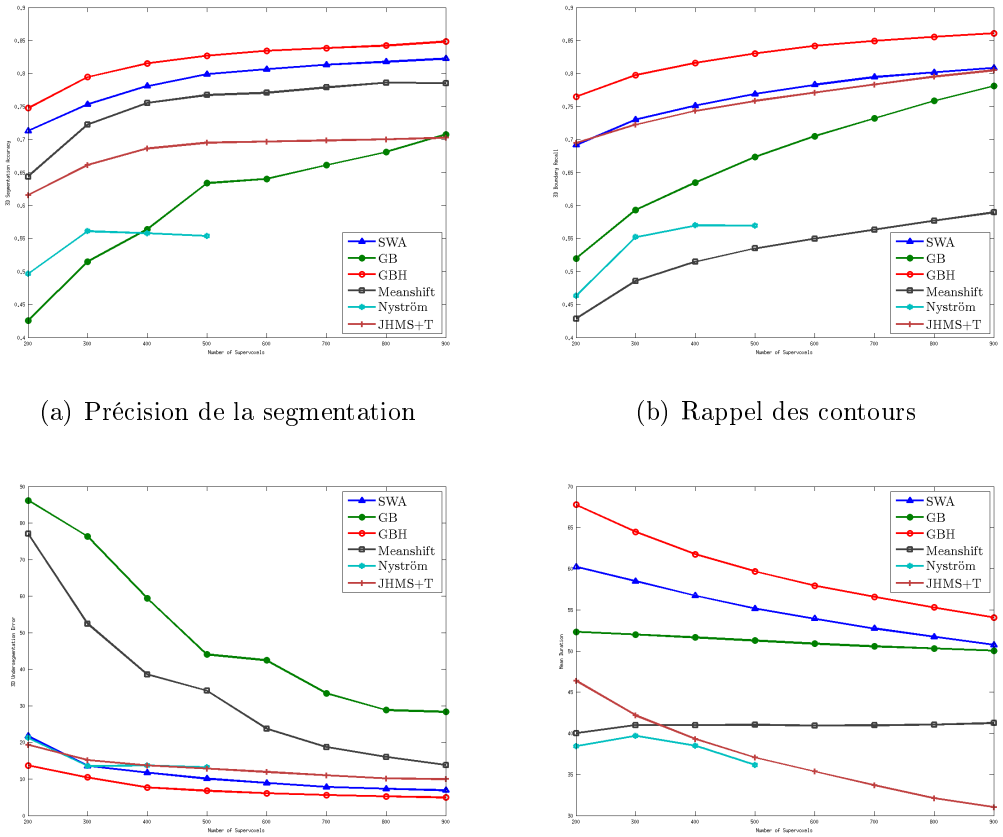
Les métriques proposées dans le benchmark reflètent la qualité d'une segmentation spatio-temporelle. En effet, la précision de la segmentation et le rappel de contours indique la qualité des régions et les contours et aide à trouver le compromis entre les deux : une bonne précision des régions ou une forte détection de contours. Cependant, la présentation des résultats ne permet pas d'identifier le meilleur jeu de paramètres : les auteurs appliquent en effet une interpolation linéaire des résultats en fonction du nombre de supervoxels. Donc pour un nombre de supervoxels donné, il est possible d'obtenir, comme dans le cas de la séquence *bus*, des résultats obtenus avec des paramètres différents que ceux utilisés pour les résultats de la séquence *container* à cause du contenu de chaque séquence.

4.3.3 Résultats visuels

Afin de valider notre approche, nous avons comparé les résultats visuels de notre segmentation avec ceux des techniques du benchmark. Afin d'avoir une comparaison juste, nous avons sélectionné des séquences segmentées avec approximativement le même nombre de supervoxels. Nous utilisons donc les seuils de fusion $T_{moy} = 20$, $T_{grad} = 10$ et $nbHier = 1$. Les valeurs de $Size_{min}$ et $Size_{max}$ dépendent de la séquence sélectionnée et sont reportées directement dans les résultats.

Dans la figure 4.12, nous comparons les résultats obtenus sur la séquence *bus*. Cette séquence est caractérisée par une forte texture se trouvant dans le bas des images (arbres, grillage) et un mouvement de la caméra pour suivre le bus. La cohérence temporelle est assurée sur toutes les segmentations. Dans notre technique, les régions en arrière-plan (arbres, mur) sont bien suivies. Le grillage aussi est suivi avec toutefois une perte de précision à cause de la fusion avec l'arrière-plan (voiture), qui est de la même couleur. Cependant, et pour toutes les techniques, le bus change d'étiquette à chaque fois qu'il passe derrière un poteau : ceci est dû au fait que ce poteau coupe la connexité des régions du bus et donc de nouvelle régions sont créées.

Dans la figure 4.13, les résultats de la segmentation *garden* sont affichés. Ici les résultats sont très intéressants. En effet, on obtient des régions de grande taille, et donc leur suivi est bien assuré (arbre, maison, ...). On remarque que du début vers la fin, l'arbre conserve bien son étiquette de départ. De plus, le ciel et les régions correspondant aux fleurs conservent bien leurs étiquettes. On remarque que les zone texturées (fleurs) sont mieux représentées que les régions



(a) Précision de la segmentation

(b) Rappel des contours

(c) Erreur de sous-segmentation

(d) Durée moyenne

Figure 4.11 – Résultats objectifs par séquence d'images

dans SWA où les régions sont découpées en plusieurs petites régions.

La dernière comparaison se fait sur la séquence *soccer* (cf. figure 4.14). Cette séquence est caractérisée par des régions faiblement texturées (terrain de foot). Ainsi, même si le sol est visuellement homogène, dans notre technique, il n'est pas considéré ici comme une seule région à cause de la contrainte de la taille. Par ailleurs, on remarque que le joueur à droite est bien suivi du début jusqu'à la fin.

En général, la technique de segmentation proposée dans ce chapitre permet de fournir des résultats visuels très intéressants. Les régions extraites sont conformes au contenu de l'image. De plus, les contours sont préservés le long de la séquence. Quant à la cohérence temporelle, les régions (arbres, joueurs) préservent leurs

étiquettes du début jusqu'à la fin de la séquence.

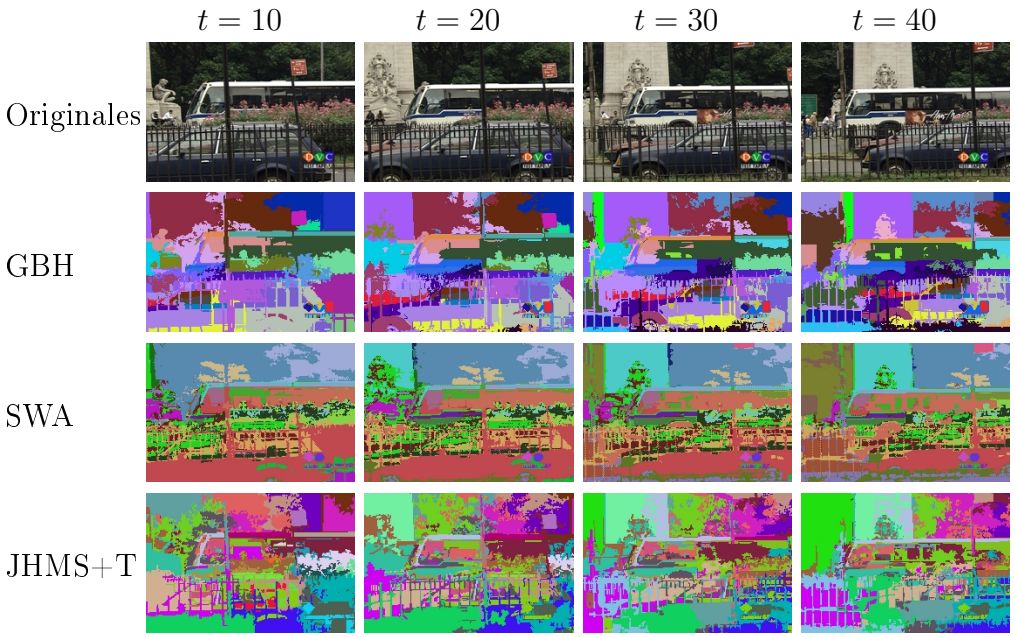


Figure 4.12 – Comparaison des segmentations sur la séquence *bus* : GBH (73 supervoxels), SWA(50 supervoxels), et notre technique JHMS+T (61 supervoxels) avec $Size_{max} = 2750$ et $Size_{min} = 275$.

4.3.4 Segmentation des images de couloir

Nous avons appliqué notre technique de segmentation sur les séquences tournées dans des couloirs. Ces séquences représentent différents couloirs avec des peintures différentes, des conditions de luminosité différentes. Comme pour la segmentation d'image (chapitre 3), nous avons été confrontés au problème du manque de texture dans les images. De plus, des régions qui représentent des objets sémantiques différents partagent les mêmes couleurs et sont seulement séparées par un contour de couleur différente. Ceci nous a contraints à réduire la valeur des seuils afin de préserver au mieux les contours des régions. Les résultats obtenus sont présentés dans la figure 4.15. Ces segmentations sont obtenues avec 4 niveaux de hiérarchie, les seuils de fusion du premier niveau de hiérarchie 1 sont $T^1_{moy} = 2$ et $T^1_{grad} = 1$. Ces deux seuils s'accroissent à chaque niveau hiérarchie de telle sorte que $T^j_{moy} = T^{j-1}_{moy} + T^1_{moy}$ pour $j = \{2...4\}$.

Dans, la première ligne de la figure 4.15, la caméra se trouve dans un petit hall menant à un couloir. Le fauteuil effectue une petite rotation avant de se

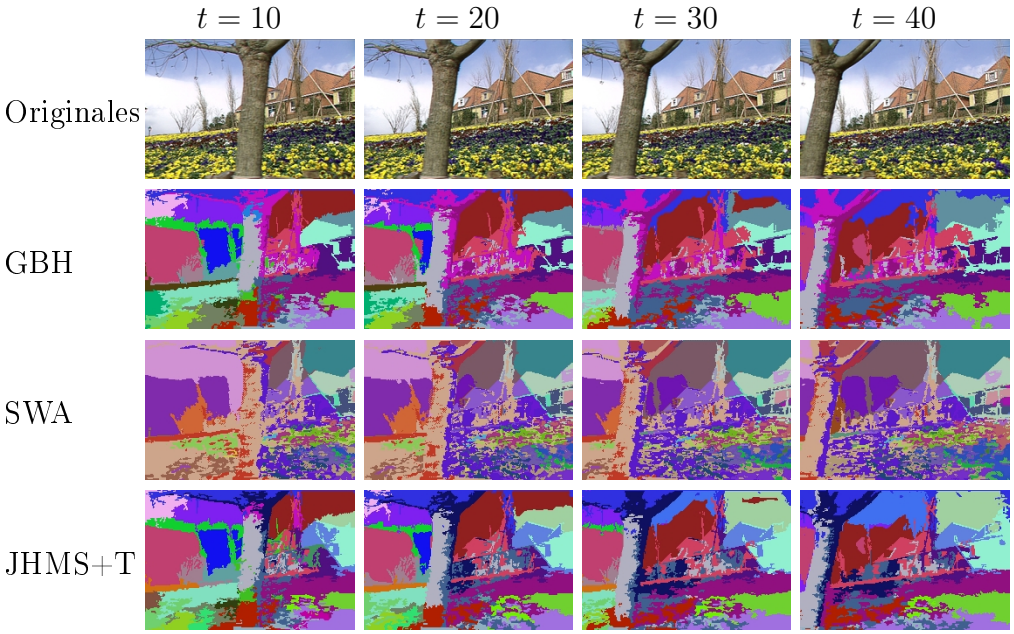


Figure 4.13 – Comparaison des segmentations sur la séquence *garden* : GBH (87), SWA (72) et notre technique JHMS+T(74 supervoxels) avec $Size_{max} = 3500$ et $Size_{min} = 350$

diriger vers le couloir. On remarque que les régions sont bien conservées (sol, mur, éléments sur le mur). Les deux lignes suivantes représentent un fauteuil qui avance en ligne droite dans un couloir. Ici, les régions se trouvant sur les murs avancent vers les bords gauches et droits de l'image.

On remarque que pour toutes les séquences, le sol est bien détecté, et son suivi est bien assuré puisque il est de couleur différente du le reste de la scène. Dans ce type de séquence, on a toutefois remarqué la difficulté à suivre les régions qui s'étendent verticalement, ces régions correspondent généralement aux montants des portes : l'absence de recouvrement lors d'un mouvement horizontal implique une perte de l'étiquette.

Lorsque des régions de même couleur sont séparées seulement par un contour de fort gradient, on remarque une perte dans la précision des contours (cf figure 4.16). En effet, les pixels des zones de mouvement obtenus après la projection des contours partagent les mêmes informations de luminance : ces pixels sont alors fusionnés avec la première région qui se présente lors de l'étape de raffinement de la segmentation. Cette perte de précision est due donc à l'ordre de fusion qui est difficilement contrôlable dans ce cas.

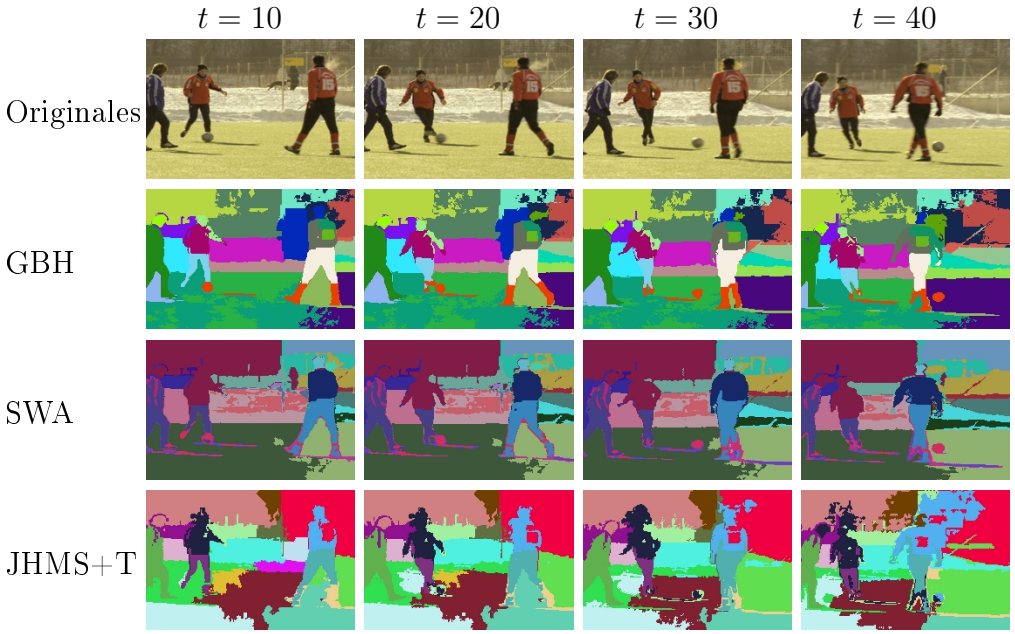


Figure 4.14 – Comparaison des segmentations sur la séquence *soccer* : GBH (28), SWA (30)et notre technique JHMS+T (28 supervoxels) avec $Size_{max} = 5000$ et $Size_{min} = 500$.

4.4 Conclusion

Dans ce chapitre, nous avons présenté un schéma général de segmentation vidéo basé sur la projection des contours. C'est une technique 2D+T qui exploite les mouvements instantanés entre deux images successives pour suivre les régions tout au long d'une séquence.

Le suivi de contours est effectué par une projection des points de contours suivant un algorithme KLT. Pour chaque contour projeté, on définit une zone de mouvement qui correspond aux changements dans l'image induits par le mouvement. Afin de détecter des nouvelles régions issues du mouvement de la caméra, nous appliquons une opération de différence de quadtree entre deux images successives. Cette opération permet de détecter les blocs homogènes de l'image précédente qui ont été décomposés par l'apparition d'un nouveau contour issu d'une nouvelle région. Ces blocs sont ajoutés aux zones de mouvement afin de reconsidérer leur appartenance à de nouvelles régions. Les zones de mouvement sont soit fusionnées avec les régions de départ afin de raffiner la segmentation de l'image précédente, soit regroupées pour former de nouvelles régions.

Les résultats obtenus sur les séquences naturelles sont très intéressants. Les

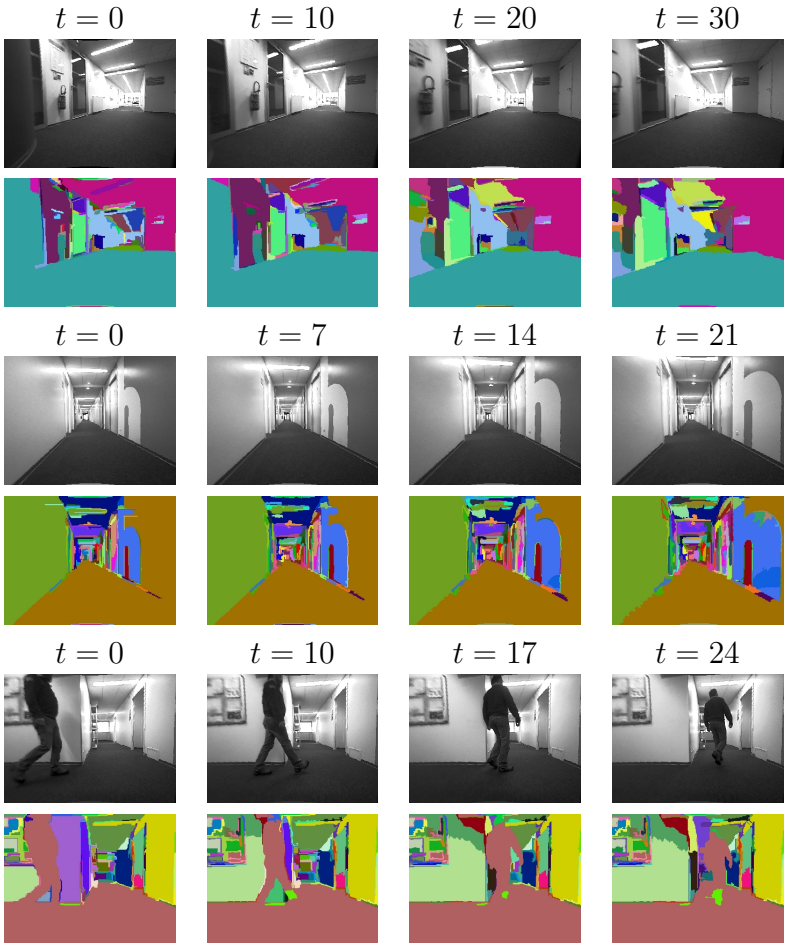


Figure 4.15 – Segmentations sur les séquences *couloir*

scores objectifs de notre algorithme sont quasiment comparables aux techniques 3D qui considèrent la vidéo comme un volume 3D. Grâce à la projection des contours, les zones sont bien conservées et leurs contours aussi précis. Visuellement, la cohérence spatio-temporelle des régions est bien assurée dans la séquence. Cependant, en termes de durée des régions, la technique souffre d'une durée moyenne de régions faible lorsque les régions sont de petites tailles et que le mouvement est important. L'absence de recouvrement est la source de cette observation.

Les tests sur les séquences type *couloir* montrent qu'il est possible d'obtenir des résultats intéressants à des seuils de fusion très bas. Cependant, la technique est sensible aux conditions d'illumination. Sachant que les régions sont de même couleur, des débordements de contours interviennent à cause du manque de gra-

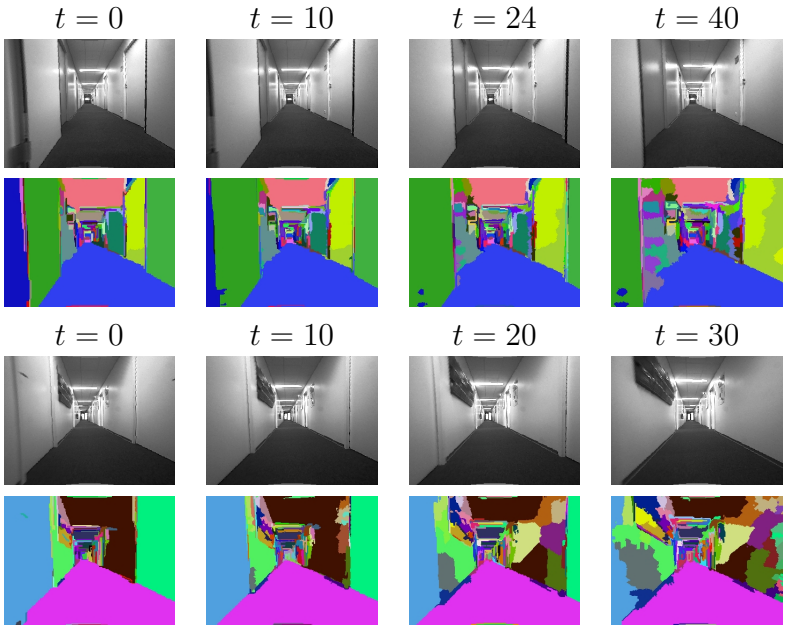


Figure 4.16 – Échec de segmentation (débordement de régions à cause de la perte des contours)

dient.
Même si en termes de représentation sémantique, la segmentation fournit un ensemble de régions cohérent avec le contenu de la scène, son utilisation à des fins de navigation reste toutefois difficile à envisager à cause de la complexité : la segmentation spatio-temporelle proposée ne répond pas au critère de temps-réel pour des images de résolution 320×240. Des optimisations devront être faites afin d'accélérer le traitement de la segmentation si on souhaite l'intégrer dans le module du fauteuil.

Conclusion

Ce mémoire de thèse propose un ensemble de contributions dans le domaine de la détection et le suivi d'objet 2D. Dans ce contexte, nous avons abordé deux problématiques principales. La première concerne la détection et le suivi des portes dans un environnement intérieur. La seconde consiste à extraire et à suivre des régions pseudo-sémantiques dans une séquence d'images. Ces solutions sont destinées à enrichir le système d'assistance à la navigation embarquée sur un fauteuil roulant électrique développé par l'équipe Lagadic (projet APASH).

Afin de pouvoir offrir à l'utilisateur du fauteuil roulant la possibilité d'une assistance lors de franchissement d'une porte, il faut être capable de connaître la position de cette porte (détection) et son déplacement dans la scène (suivi). Les deux premiers chapitres (1 et 2) ont donc porté respectivement sur des solutions de détection et de suivi de portes. Les solutions proposées sont purement basée vision et utilisent une caméra monoculaire embarquée directement sur le fauteuil. La détection des portes repose sur un ensemble de descripteurs géométriques et les connaissances à priori de la scène. Le principe de la détection est basé sur l'identification des formes rectangulaires dans l'image respectant les normes standards d'une porte traduites dans une perspective en profondeur. Cette recherche de formes s'effectue dans des zones correspondant aux murs dans l'image. Ces zones sont délimitées à travers la détection des lignes séparatrices sol/mur. Au fur et à mesure de la navigation, la forme de la porte dans l'image subit plusieurs transformations dues à la rotation et à la distance par rapport à la caméra. Afin de pouvoir récupérer ces paramètres, nous avons proposé une technique de localisation de la caméra dans un couloir qui permet de retrouver sa position relative ainsi que sa rotation dans le couloir. Enfin, la phase de détection de la porte consiste à identifier deux montants dont la distance respecte la largeur réelle de la porte transformée dans le plan image. Les résultats obtenus montrent que l'algorithme est capable de détecter plus de 82% de portes. Cette technique a montré sa capacité à détecter des portes ouvertes ou fermées, des portes de la même couleur que le mur et même des portes à des distances importantes.

Ensuite, une fois qu'une porte est détectée, on initialise son suivi afin de

connaître son déplacement dans l'image. Nous avons basé le suivi de la porte sur celui de ses deux montants qui sont caractérisés par un fort gradient. Chaque montant est suivi à travers une technique de suivi de ligne 2D. Cette approche est caractérisée par sa rapidité : les portes sont suivies en temps réel et les résultats montrent la robustesse de la méthode vis-à-vis des changements d'illumination. Le schéma global de détection et de suivi offre une solution complète et automatique qui peut être embarquée dans le système de navigation du fauteuil roulant.

Ensuite, nous avons abordé une autre problématique dans le troisième chapitre : l'extraction d'une représentation pseudo-sémantique d'une image. Son but est de proposer une représentation interprétable par un système de navigation pour la réalisation de différentes tâches (détection de zones navigables, détection d'obstacles...). Cette extraction repose sur une technique de segmentation d'images qui permet de regrouper les zones homogènes dans l'image. Nous avons proposé dans ce chapitre une nouvelle technique de segmentation dénommée JHMS et caractérisée par les propriétés de scalabilité en résolution et en hiérarchie. Pour ce faire, nous avons introduit la notion du graphe d'adjacence (RAG) multirésolution. Ce RAG permet de projeter les régions d'une résolution à une autre en permettant le raffinement de la segmentation à mesure qu'on descend dans la pyramide. De plus, à chaque niveau du RAG, une segmentation hiérarchique est effectuée afin de fournir une représentation de la scène ajustable. Les résultats obtenus sont très intéressants. D'une part, les scores objectifs montrent que notre technique est comparable à ses concurrentes : elle se classe en deuxième position en termes de qualité des régions extraites par rapport à la vérité terrain et quatrième en termes de qualité de ses contours. D'autre part, les résultats visuels sont aussi satisfaisants : les régions et les contours sont en concordance avec le contenu de l'image.
Afin d'appliquer la segmentation sur des images de type couloir, nous avons adapté les paramètres nécessaire à l'algorithme en fonction des caractéristiques de la scène à savoir un faible gradient et un manque de texture. Les résultats visuels ont montré la capacité de la segmentation à extraire une bonne représentation en régions (sol, murs...).

Enfin, nous avons présenté une extension du JHMS pour la segmentation spatio-temporelle dans le dernier chapitre. L'objectif est de fournir une représentation pseudo-sémantique d'une vidéo en assurant une cohérence spatio-temporelle des régions. Pour ce faire, nous avons proposé une technique de segmentation 2D+T basée sur la projection des contours d'une image à une autre. La projection des contours permet de cibler les zones de mouvements affectées par le déplacement des régions. Ces contours sont projetés en utilisant l'algorithme KLT et corrigés à travers une modélisation par mélange de gaussiennes. Nous avons

aussi appliqué une technique de différence entre deux quadtrees successifs afin de détecter les nouvelles régions qui apparaissent entre les deux images correspondantes. Un raffinement de la segmentation est effectué à la fin pour fusionner les régions projetées avec les zones de mouvement et les nouvelles régions obtenues. Les résultats objectifs sont très intéressants là encore : les scores obtenus sur la qualité des régions ainsi que les contours détectés sont comparables avec ceux des techniques 3D. Cependant, la cohérence temporelle est faible comparée à celle des techniques 3D à cause la contrainte de taille des régions qui réduit le taux de recouvrement des régions entre deux images successives. Les résultats visuels montrent que la représentation en régions est conforme avec le contenu de la séquence. En outre, la cohérence temporelle dépend du contenu de chaque séquence : plus la séquence est texturée plus la cohérence est faible, ce qui est évident puisque les descripteurs utilisés sont la couleur et le gradient.

Perspectives

Une bonne détection de porte repose sur l'efficacité de l'extracteur de lignes dans l'image. Nous avons opté pour l'algorithme LSD basé sur l'orientation des gradients. On a montré que le problème avec cette technique reste la décomposition des segments lorsqu'il y a un changement d'orientation de gradient. Nous avons corrigé cela en ajoutant une étape de fusion de lignes. L'utilisation d'une autre alternative peut s'avérer utile pour pallier ce problème [Jang 02] qui consiste à regrouper des contours obtenus à travers un détecteur de contours dans le but de reconstituer des segments en fonction de l'angle et la distance.

De plus, l'objectif est d'étendre le schéma de détection et de suivi des portes pour d'autres types environnement (chambre, bureau, hall...) où les points de fuite ne sont plus visibles dans l'image. Dans ces types d'environnement, on peut retrouver plusieurs éléments d'intérieur (table, chaises...) qui peuvent altérer la détection des lignes sol/mur. Il faudra trouver donc d'autres moyens pour la localisation du fauteuil afin de détecter les portes. Pour cela, on peut se baser sur les lignes du plafond plutôt que celle du sol afin d'estimer la position du mur par rapport à la caméra.

La segmentation d'images repose sur un ensemble de critères pour la fusion des régions. Lors de nos expérimentations, nous avons cherché l'ensemble des paramètres qui offrent les meilleurs résultats en appliquant une recherche exhaustive sur les combinaisons de valeurs de paramètres possibles. Cette combinaison est pertinente pour des images naturelles mais elle n'est plus vraie pour d'autres types d'images (images de couloir en particulier). L'adaptation des seuils en fonction du type d'images est une étape très importante pour extraire une bonne représentation en fonction du contenu de l'image. Cette adaptation

devrait pouvoir s'effectuer automatiquement en fonction du contenu de l'image (distribution des couleurs, histogramme des gradients, texture ...).

Par ailleurs, le processus de fusion ne permet pas de contrôler l'ordre de fusion des régions. Pour pallier ce problème, il serait judicieux de définir un classement des liens dans le RAG en fonction des similarités entre deux régions voisines, avant de la fusion dans l'ordre des régions les plus similaires. Cette technique peut s'avérer toutefois coûteuse, puisque à chaque itération, il faut effectuer un tri des liens du RAG qui peuvent être très nombreux.

Enfin, les images segmentées sont représentées ici dans l'espace de couleur YUV. Cependant, des techniques utilisent classiquement le L*a*b comme espace de couleur pour segmenter les images [HernandezGomez 09]. Cet espace fourni des différences de couleurs proches de celles perçues par le système visuel humain. Il serait donc intéressant de tester le JHMS en utilisant cet espace couleur en définissant des seuils de fusion adaptés [Strauss 10].

En ce qui concerne la segmentation spatio-temporelle, pour la projection des contours, nous avons utilisé le KLT pour l'estimation du mouvement des point de contours entre deux images successives. Il serait intéressant d'appliquer plutôt un suivi de contour sur leur normale (comme pour le suivi de portes) qui serait plus adapté à notre application. Par ailleurs, la projection des contours nous a permis de définir des zones de mouvements. Cependant, nous n'avons pas utilisé les informations de déplacement des contours pour projeter les pixels des régions. Il serait donc intéressant d'exploiter le mouvement des contours pour projeter l'ensemble des pixels des régions. En appliquant cette solution, on supprimerait les zones de mouvement et le raffinement de la segmentation reviendrait à simplement regrouper les nouvelles régions issues de la différence des quadtrees.

Bibliographie

[Albuquerque 04] de M. Portes Albuquerque, I. A. Esquef, A. R. Gesualdi Mello. – Image thresholding using tsallis entropy. Pattern Recogn. Lett., 25(9) :1059–1065, juillet 2004.

[AlNajdawi 12] Nijad Al-Najdawi, Sara Tedmori, Eran A Edirisinghe, Helmut E Bez. – An automated real-time people tracking system based on klt features detection. Int. Arab J. Inf. Technol., 9(1) :100–107, 2012.

[Alvarez 10] Luis Alvarez, Luis Gomez, J. Rafael Sendra. – Algebraic Lens Distortion Model Estimation. Image Processing On Line, 2010, 2010.

[Aneja 09] K. Aneja, F. Laguzet, L. Lacassagne, A. Merigot. – Video-rate image segmentation by means of region splitting and merging. – Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on, pp. 437–442, 2009.

[Anguelov 04] D. Anguelov, D. Koller, E. Parker, S. Thrun. – Detecting and modeling doors with mobile robots. – IEEE International Conference on Robotics and Automation, vol. 4, pp. 3777 – 3784, 2004.

[Arbelaez 11] Pablo Arbelaez, Michael Maire, Charless Fowlkes, Jitendra Malik. – Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 33(5) :898–916, mai 2011.

[Babel 12] M. Babel. – From image coding and representation to robotic vision. – Habilitation à diriger les recherches, Université Rennes 1, June 2012.

[Baptiste 13] Brun Baptiste. – Détection de protes. – Université de Rennes 1, janvier 2013.

[Bazin 12] J. C Bazin, Yongduek Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, Inso Kweon, M. Pollefeys. – Globally optimal

line clustering and vanishing point estimation in manhattan world. – Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 638–645, 2012.

[Benhimane 07] S. Benhimane, E. Malis. – Homography-based 2d visual tracking and servoing. Int. J. Rob. Res., 26(7) :661–676, juillet 2007.

[Berger 94] Marie-Odile Berger. – How to track efficiently piecewise curved contours with a view to reconstructing 3d objects. – In Int. Conf on Pattern Recognition, ICPR94, pp. 32–36, 1994.

[Bertolino 96] P. Bertolino, Annick Montanvert. – Multiresolution segmentation using the irregular pyramid. – Image Processing, 1996. Proceedings., International Conference on, vol. 1, pp. 257–260 vol.1, 1996.

[Bilmes 98] Jeff Bilmes. – A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. – Rapport de recherche, 1998.

[Bins 09] J. Bins, C.R. Jung, L.L. Dihl, A. Said. – Feature-based face tracking for videoconferencing applications. – Multimedia, 2009. ISM '09. 11th IEEE International Symposium on, pp. 227–234, 2009.

[Bister 90] M. Bister, J. Cornelis, A. Rosenfeld. – A critical view of pyramid segmentation algorithms. Pattern Recognition Letters, 11(9) :605 – 617, 1990.

[Boukir 98] Samia Boukir, Patrick Bouthemy, FranÃ§ois Chaumette, Didier Juvin. – A local method for contour matching and its parallel implementation. Machine Vision and Applications, 10(5-6) :321–330, 1998.

[Boulanger 06] Kadi Boulanger, Kadi Bouatouch, Sumant Pattanaik. – ATIP : A Tool for 3D Navigation inside a Single Image with Automatic Camera Calibration. – EUROGRAPHICS (édité par), EG UK conference, Middlesbrough, Royaume-Uni, juin 2006. EUROGRAPHICS, WILEY.

[Bouthemy 89] P. Bouthemy. – A maximum likelihood framework for determining moving edges. IEEE trans on Pattern Analysis and Machine Intelligence, 11(5) :499–511, May 1989.

[Braviano 95] Braviano, Gilson. – Logique floue en segmentation d'images : Seuillage par entropie et structures pyramidales

irrégulières. – PhD. Thesis, Université Joseph Fourrier-Grenoble 1, 10 1995.

[Brox 11] T. Brox, J. Malik. – Large displacement optical flow : Descriptor matching in variational motion estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(3) :500–513, 2011.

[Brun 02] Luc Brun. – Traitement d'images couleur et pyramides combinatoires. – Habilitation à diriger des recherches, Université de Reims Champagne-Ardenne, France, décembre 2002.

[Burt 81] Peter J Burt. – Fast filter transform for image processing. Computer Graphics and Image Processing, 16(1) :20 – 51, 1981.

[Burt 83] P.J. Burt, E.H. Adelson. – The laplacian pyramid as a compact image code. Communications, IEEE Transactions on, 31(4) :532–540, 1983.

[Calderero 10] Felipe Calderero, Ferran Marques. – Region merging techniques using information theory statistical measures. Trans. Img. Proc., 19(6) :1567–1586, juin 2010.

[Canny 86] John Canny. – A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6) :679–698, 1986.

[Cantoni 86] Virginio Cantoni, Stefano Levialdi (édité par). – Pyramidal systems for computer vision. – Springer-Verlag, London, UK, UK, 1986.

[Cao 11] Xianbin Cao, Jinhe Lan, Pingkun Yan, Xuelong Li. – Klt feature based vehicle detection and tracking in airborne videos. – Image and Graphics (ICIG), 2011 Sixth International Conference on, pp. 673–678, 2011.

[Carinena 04] P. Carinena, C.V. Regueiro, A. Otero, A.J. Bugarin, S. Barro. – Landmark detection in mobile robotics using fuzzy temporal rules. Fuzzy Systems, IEEE Transactions on, 12(4) :423–435, 2004.

[Castagno 98] R. Castagno, T. Ebrahimi, M. Kunt. – Video segmentation based on multiple features for interactive multimedia applications. Circuits and Systems for Video Technology, IEEE Transactions on, 8(5) :562–571, 1998.

[CCITT 92] CCITT. – Information technology digital - compression and coding of continuous-tone still images - requirements and guidelines, iso/iec 10918 1, itu t.81. 1992.

[Celenk 90] Mehmet Celenk. – A color clustering technique for image segmentation. Computer Vision, Graphics, and Image Processing, 51(3) :370, 1990.

[Chen 80] P.C. Chen, T. Pavlidis. – Image segmentation as an estimation problem. Computer Graphics and Image Processing, 12(2) :153 – 172, 1980.

[Chen 08] Zhichao Chen, S.T. Birchfield. – Visual detection of lintel-occluded doors from a single image. – Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on, pp. 1–8, 2008.

[Chen 10] A.Y.C. Chen, J.J. Corso. – Propagating multi-class pixel labels throughout video frames. – Image Processing Workshop (WNYIPW), 2010 Western New York, pp. 14–17, 2010.

[Cheng 00] Heng-Da Cheng, Ying Sun. – A hierarchical approach to color image segmentation using homogeneity. Image Processing, IEEE Transactions on, 9(12) :2071–2082, 2000.

[Cheng 07] Dong Seon Cheng, M. A T Figueiredo. – Cosegmentation for image sequences. – Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on, pp. 635–640, 2007.

[Colantoni 97] P. Colantoni, B. Laget. – Color image segmentation using region adjacency graphs. – Image Processing and Its Applications, 1997., Sixth International Conference on, vol. 2, pp. 698–702 vol.2, 1997.

[Collins 89] Robert Collins, R. Weiss. – An efficient and accurate method for computing vanishing points. – Topical Meeting on Image Understanding and Machine Vision, vol. 14, pp. 92–94, Washington, D.C, 1989. Optical Society of America.

[Comaniciu 02] D. Comaniciu, P. Meer. – Mean shift : a robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(5) :603–619, 2002.

[Corso 08] **J. J. Corso**, E. Sharon, S. Dube, S. El-Saden, U. Sinha, A. Yuille. – Efficient Multilevel Brain Tumor Segmenta-

tion with Integrated Bayesian Model Classification. IEEE Transactions on Medical Imaging, 27(5) :629–640, 2008.

[Coutard 11] L. Coutard, F. Chaumette. – Visual detection and 3d model-based tracking for landing on aircraft carrier. – IEEE Int. Conf. on Robotics and Automation, ICRA'11, pp. 1746–1751, Shanghai, China, May 2011.

[Cui 08] Weihong Cui, Zequn Guan, Zhiyi Zhang. – An improved region growing algorithm for image segmentation. – Computer Science and Software Engineering, 2008 International Conference on, vol. 6, pp. 93–96, 2008.

[Dame 10] A. Dame, E. Marchand. – Accurate real-time tracking using mutual information. – IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'10, pp. 47–56, Seoul, Korea, October 2010.

[Delage 06] E. Delage, Honglak Lee, A.Y. Ng. – A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. – , IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2418 – 2428, 2006.

[DelBimbo 00] A. Del Bimbo, P. Pala, L. Tanganelli. – Video retrieval based on dynamics of color flows. – Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 1, pp. 851–854 vol.1, 2000.

[Dementhon 95] Daniel F Dementhon, Larry S Davis. – Model-based object pose in 25 lines of code. International journal of computer vision, 15(1-2) :123–141, 1995.

[Dementhon 02] Daniel Dementhon. – Spatio-temporal segmentation of video by hierarchical mean shift analysis. – Center for Automat. Res., U. of Md, College Park, 2002.

[Deng 98] Yining Deng, B.S. Manjunath. – Spatio-temporal relationships and video object extraction. – Signals, Systems amp ; Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on, vol. 1, pp. 895–899 vol.1, 1998.

[Deng 01] Yining Deng, B.S. Manjunath. – Unsupervised segmentation of color-texture regions in images and video. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(8) :800–810, 2001.

[DeSimone 08] Francesca De Simone, Daniele Ticca, Frederic Dufaux, Michael Ansorge, Touradj Ebrahimi. – A comparative study of color image compression standards using perceptually driven quality metrics. – Optical Engineering+ Applications, pp. 70730Z–70730Z. International Society for Optics and Photonics, 2008.

[Déforges 07] O. Déforges, M. Babel, L. Bédat, J. Ronsin. – Color LAR codec : a color image representation and compression scheme based on local resolution adjustment and self-extracting region representation. IEEE Transactions on Circuits and Systems for Video Technology, 17(8) :974–987, 2007.

[Dollar 06] P. Dollar, Zhuowen Tu, S. Belongie. – Supervised learning of edges and object boundaries. – Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, pp. 1964–1971, 2006.

[Dorea 07] C.C. Dorea, M. Pardas, F. Marques. – Hierarchical partition-based representations for image sequences using trajectory merging criteria. – Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol. 1, pp. I–1077–I–1080, 2007.

[Dowson 08] Nicholas Dowson, Richard Bowden, Senior Member. – Mutual information for lucas-kanade tracking (milk) : An inverse compositional formulation. – In IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008.

[Duda 72] Richard O Duda, Peter E Hart. – Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1) :11–15, 1972.

[Duda 73] Richard O Duda, Peter E Hart et al. – Pattern classification and scene analysis. – Wiley New York, vol. 3, 1973.

[Dufaux 95] F. Dufaux, F. Moscheni, A. Lippman. – Spatio-temporal segmentation based on motion and static segmentation. – Image Processing, 1995. Proceedings., International Conference on, vol. 1, pp. 306–309 vol.1, 1995.

[Elder 96] James H Elder, Steven W Zucker. – Computing contour closure. Computer Vision ECCV96, pp. 399–412. – Springer, 1996.

[ElKaissi 07] M. ElKaissi, M. Elgamel, M. Bayoumi, B. Zavidovique. – Sedlrf : A new door detection system for topological

maps. – Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on, pp. 75–80, 2007.

[Felzenszwalb 04] Pedro F. Felzenszwalb, Daniel P. Huttenlocher. – Efficient graph-based image segmentation. Int. J. Comput. Vision, 59(2) :167–181, septembre 2004.

[Fischler 81] Martin A. Fischler, Robert C. Bolles. – Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24(6) :381–395, juin 1981.

[Foret 02] G. Foret, P. Bertolino, D. Cibaud. – Partition projection in videos by global and local block-matching. – Image Processing. 2002. Proceedings. 2002 International Conference on, vol. 3, pp. III–409–III–412 vol.3, 2002.

[Fowlkes 01] C. Fowlkes, S. Belongie, J. Malik. – Efficient spatiotemporal grouping using the nystrom method. – Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, pp. I–231–I–238 vol.1, 2001.

[Freixenet 02] J. Freixenet, X. Muñoz, D. Raba, J. Martí, X. Cufí. – Yet another survey on image segmentation : Region and boundary information integration. – In ECCV, pp. 408–422, 2002.

[Galmar 05] E Galmar, B Huet. – Méthode de segmentation par graphe pour le suivi de régions spatio-temporelles. Journées CORESA, 2005.

[Gambotto 93] Jean-Pierre Gambotto. – A new approach to combining region growing and edge detection. Pattern Recogn. Lett., 14(11) :869–875, novembre 1993.

[Gevers 90] T Gevers, FCA Groen. – Segmentation of color images. 1990.

[Glasner 11] Daniel Glasner, Shiv N Vitaladevuni, Ronen Basri. – Contour-based joint clustering of multiple segmentations. – Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 2385–2392. IEEE, 2011.

[Gomila 01] Cristina Gomila. – Mise en correspondance de partitions en vue du suivi d'objets. – Paris, PhD. Thesis, Ecole nationale supérieure des Mïnes, 2001.

[Gosda 10] U. Gosda, T. Schiekel, U. Petersohn. – A unified approach to active contour snakes. – Intelligent Computer Communication and Processing (ICCP), 2010 IEEE International Conference on, pp. 171–178, 2010.

[Greenspan 02] Hayit Greenspan, Jacob Goldberger, Arnaldo Mayer. – A probabilistic framework for spatio-temporal video representation and indexing. – in IEEE Conf. on Computer Vision and Pattern Recognition, pp. 461–475. Springer-Verlag, 2002.

[Grosky 83] William I. Grosky, Ramesh Jain. – Optimal quadtrees for image segments. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-5(1) :77–83, 1983.

[Gross 87] Ari David Gross, Azriel Rosenfeld. – Multiresolution object detection and delineation. Computer Vision, Graphics, and Image Processing, 39(1) :102 – 115, 1987.

[Grundmann 10] M. Grundmann, V. Kwatra, Mei Han, I. Essa. – Efficient hierarchical graph-based video segmentation. – Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2141–2148, 2010.

[Guo 03] Chen Guo. – The fisher criterion function method of image thresholding. Chinese Journal of Scientific Instrument, 24(6) :564–567, 2003.

[Hager 95] G Hager. – The'x-vision'system : A general-purpose substrate for vision-based robotics. – Workshop on Vision for Robotics, 1995.

[Hager 98a] G.D. Hager, P.N. Belhumeur. – Efficient region tracking with parametric models of geometry and illumination. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(10) :1025–1039, 1998.

[Hager 98b] Gregory D. Hager, Kentaro Toyama. – X vision : A portable substrate for real-time vision applications. Computer Vision and Image Understanding, 69(1) :23 – 37, 1998.

[Haralick 73] R.M. Haralick, K. Shanmugam, Its'Hak Dinstein. – Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on, SMC-3(6) :610–621, 1973.

[Harris 88] Chris Harris, Mike Stephens. – A combined corner and edge detector. – In Proc. of Fourth Alvey Vision Conference, pp. 147–151, 1988.

[Hartigan 79] J. A. Hartigan, M. A. Wong. – Algorithm AS 136 : A k-means clustering algorithm. Applied Statistics, 28(1) :100–108, 1979.

[Hensler 10] J. Hensler, M. Blaich, O. Bittel. – Real-Time Door Detection Based on AdaBoost Learning Algorithm. – Research and Education in Robotics - EUROBOT 2009, Communications in Computer and Information Science, Vol. 82, p. 61, 2010.

[Herbulot 07] A. Herbulot. – Mesures statistiques non-paramétriques pour la segmentation d'images et de vidéos et minimisation par contours actifs. – 2007.

[HernandezGomez 09] G. Hernandez-Gomez, R.E. Sanchez-Yanez, V. Ayala-Ramirez, F.E. Correa-Tome. – Natural image segmentation using the cielab space. – Electrical, Communications, and Computers, 2009. CONIELECOMP 2009. International Conference on, pp. 107–110, 2009.

[Hilsmann 07] A. Hilsmann, P. Eisert. – Deformable object tracking using optical flow constraints. – Visual Media Production, 2007. IETCVMP. 4th European Conference on, pp. 1–8, 2007.

[Horn 80] Berthold K.P. Horn, Brian G. Schunck. – Determining Optical Flow. – Rapport de recherche, Cambridge, MA, USA, 1980.

[Horn 81] Berthold K. P. Horn, Brian G. Schunck. – Determining optical flow. ARTIFICAL INTELLIGENCE, 17 :185–203, 1981.

[Horn 86] Berthold K. Horn. – Robot Vision. – McGraw-Hill Higher Education, 1st édition, 1986.

[Horowitz 74] S. L. Horowitz, T. Pavlidis. – Picture Segmentation by a directed split-and-merge procedure. Proceedings of the 2nd International Joint Conference on Pattern Recognition, Copenhagen, Denmark, pp. 424–433, 1974.

[Horowitz 76] Steven L. Horowitz, Theodosios Pavlidis. – Picture segmentation by a tree traversal algorithm. J. ACM, 23(2) :368–388, avril 1976.

[Huang 06] Yu-Wen Huang, Ching-Yeh Chen, Chen-Han Tsai, Chun-Fu Shen, Liang-Gee Chen. – Survey on block matching motion estimation algorithms and architectures with new results. VLSI Signal Processing, pp. 297–320, 2006.

[Inaki 02] MONASTERIO Inaki, LAZKANO Elena, RANO Inaki, SIERRA Basilo, TZAFESTAS Spyros G., TZAFESTAS Elpida S. – Learning to traverse doors using visual information. Mathematics and computers in simulation, 60(3-5) :347–356, 2002. – eng.

[Ince 08] S. Ince, J. Konrad. – Occlusion-aware optical flow estimation. Image Processing, IEEE Transactions on, 17(8) :1443–1451, 2008.

[Jang 02] Jeong-Hun Jang, Ki-Sang Hong. – Fast line segment grouping method for finding globally more favorable line segments. Pattern Recognition, 35(10) :2235 – 2247, 2002.

[Jiang 00] X. Jiang. – An adaptive contour closure algorithm and its experimental evaluation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(11) :1252–1265, 2000.

[KaraFalah 94] R. Kara Falah, P. Bolon, J-P Cocquerez. – A region-region and region-edge cooperative approach of image segmentation. – Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, vol. 3, pp. 470–474 vol.3, 1994.

[Kass 88] Michael Kass, Andrew Witkin, Demetri Terzopoulos. – Snakes : Active contour models. International Journal of Computer Vision, 1(4) :321–331, 1988.

[Ke 02] Qifa Ke, Takeo Kanade. – A robust subspace approach to layer extraction, 2002.

[Kim 94] Dongsung Kim, , Dongsung Kim, Ramakant Nevatia. – A method for recognition and localization of generic objects for indoor navigation, 1994.

[Kim 11] Soohwan Kim, Howon Cheong, Dong Hwan Kim, Sung-Kee Park. – Context-based object recognition for door detection. – Advanced Robotics (ICAR), 2011 15th International Conference on, pp. 155–160, 2011.

[Li 09] Yan Li, Lei Li. – A split and merge em algorithm for color image segmentation. – Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on, vol. 4, pp. 395–399, 2009.

[Lucas 81a] Bruce D. Lucas, Takeo Kanade. – An iterative image registration technique with an application to stereo vision. – pp. 674–679, 1981.

[Lucas 81b] Bruce D. Lucas, Takeo Kanade. – An iterative image registration technique with an application to stereo vision. – Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, IJCAI'81, pp. 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[Maire 08] Michael Maire, Pablo Arbelaez, Charless Fowlkes, Jitendra Malik. – Using contours to detect and localize junctions in natural images. – CVPR, 2008.

[Marchand 05] E. Marchand, F. Chaumette. – Feature tracking for visual servoing purposes, 2005.

[Marfil 06] R. Marfil, L. Molina-Tanco, A. Bandera, J. A. Rodríguez, F. Sandoval. – Pyramid segmentation algorithms revisited. Pattern Recogn., 39(8) :1430–1451, août 2006.

[Marques 98] F. Marques, J. Llach. – Tracking of generic objects for video object generation. – Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, pp. 628–632 vol.3, 1998.

[Martin 01] D. Martin, C. Fowlkes, D. Tal, J. Malik. – A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. – Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 2, pp. 416–423 vol.2, 2001.

[Martin 04] D.R. Martin, C.C. Fowlkes, J. Malik. – Learning to detect natural image boundaries using local brightness, color, and texture cues. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(5) :530–549, 2004.

[Meer 89] Peter Meer. – Stochastic image pyramids. Comput. Vision Graph. Image Process., 45(3) :269–294, mars 1989.

[Megret 02] Remi Megret, Daniel Dementhon. – A Survey of Spatio-Temporal Grouping Techniques. – Rapport de recherche, 2002.

[Merigot 03] A. Merigot. – Revisiting image splitting. – Image Analysis and Processing, 2003.Proceedings. 12th International Conference on, pp. 314–319, 2003.

[Mittal 12] Ajay Mittal, Sanjeev Sofat, Edwin Hancock. – Detection of edges in color images : A review and evaluative comparison of state-of-the-art techniques. Autonomous and Intelligent

Systems, éd. par Mohamed Kamel, Fakhri Karray, Hani Hagras, pp. 250–259. – Springer Berlin Heidelberg, 2012.

[Mémin 02] Etienne Mémin, Patrick PÃ©rez. – Hierarchical estimation and segmentation of dense motion fields. International Journal of Computer Vision, 46(2) :129–155, 2002.

[Montanvert 91] Annick Montanvert, P. Meer, A. Rosenfield. – Hierarchical image analysis using irregular tessellations. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 13(4) :307–316, 1991.

[Montoya 03] Maria Dolores Gil Montoya, C. Gil, I. Garcia. – The load unbalancing problem for region growing image segmentation algorithms. Journal of Parallel and Distributed Computing, 63(4) :387 – 395, 2003.

[Morros 04] Ramon Morros. – Optimization of Segmentation Based Video Sequence Coding Technique : Application to Content Based Functionalities. – PhD. Thesis, Technical University of Catalonia (UPC), October 2004.

[Moscheni 98] F. Moscheni, S. Bhattacharjee, M. Kunt. – Spatio-temporal segmentation based on region merging. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(9) :897–915, 1998.

[MuñozSalinas 04] Rafael Muñoz-Salinas, Eugenio Aguirre, Miguel García-Silvente, Antonio González. – Door-detection using computer vision and fuzzy logic. – 6th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering, 2004.

[Murillo 08] A. C. Murillo, J. Košecká, J. J. Guerrero, C. Sagüés. – Visual door detection integrating appearance and shape cues. Robot. Auton. Syst., 56(6) :512–521, juin 2008.

[Ngo 08] Thanh Duc Ngo, Duy-Dinh Le, S. Satoh, Duc Anh Duong. – Robust face track finding in video using tracked points. – Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on, pp. 59–64, 2008.

[Ohlander 78] Ron Ohlander, Keith Price, D. Raj Reddy. – Picture segmentation using a recursive region splitting method. Computer Graphics and Image Processing, 8(3) :313–333, décembre 1978.

[Ok 12] Kyel Ok, Duy-Nguyen Ta, Frank Dellaert. – Vistas and wall-floor intersection features - enabling autonomous flight in man-made environments. – Workshop on Visual Control of Mobile Robots, 2012.

[Oliver 06] A. Oliver, X. Munoz, J. Batlle, L. Pacheco, J. Freixenet. – Improving clustering algorithms for image segmentation using contour and region information. – Automation, Quality and Testing, Robotics, 2006 IEEE International Conference on, vol. 2, pp. 315–320, 2006.

[Otsu 79] N. Otsu. – A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics, 9(1) :62–66, janvier 1979.

[Palou 13] Guillem Palou, Philippe Salembier. – Hierarchical video representation with trajectory binary partition tree. – CVPR2013, 2013.

[Panin 06] G. Panin, A. Ladikos, A. Knoll. – An efficient and robust real-time contour tracking system. – Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on, pp. 44–44, 2006.

[Pardas 94] M. Pardas, P. Salembier. – Joint region and motion estimation with morphological tools. Mathematical Morphology and Its Applications to Image Processing, éd. par Jean Serra, Pierre Soille, pp. 93–100. – Springer Netherlands, 1994.

[Paris 07] S. Paris, F. Durand. – A topological approach to hierarchical segmentation using mean shift. – Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, pp. 1–8, 2007.

[Paris 08] Sylvain Paris. – Edge-preserving smoothing and mean-shift segmentation of video streams. – Proceedings of the 10th European Conference on Computer Vision : Part II, ECCV '08, pp. 460–473, Berlin, Heidelberg, 2008. Springer-Verlag.

[Park 00] Dong Kwon Park, Ho Seok Yoon, Chee Sun Won. – Fast object tracking in digital video. Consumer Electronics, IEEE Transactions on, 46(3) :785–790, 2000.

[Paschos 01] G. Paschos. – Perceptually uniform color spaces for color texture analysis : an empirical evaluation. Image Processing, IEEE Transactions on, 10(6) :932–937, 2001.

[Pasteau 10] F. Pasteau, M. Babel, O. Déforges, C. Strauss, L. Bédat. – Locally Adaptive Resolution (LAR) codec. Recent Advances in Signal Processing, éd. par Ashraf A. Zaher, pp. 37–48. – IN-TECH Education and Publishing, November 2010.

[Pasteau 13] F. Pasteau, M. Babel, R. Sekkal. – Corridor following wheelchair by visual servoing. – IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'2013, Tokyo, Japan, November 2013.

[Petit 12] A. Petit, E. Marchand, K. Kanani. – Tracking complex targets for space rendezvous and debris removal applications. – IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'12, pp. 4483–4488, Vilamoura, Portugal, October 2012.

[Pletzer 11] F. Pletzer, R. Tusch, L. Boszormenyi, B. Rinner, O. Sidla, M. Harrer, T. Mariacher. – Feature-based level of service classification for traffic surveillance. – Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pp. 1015–1020, 2011.

[PontTuset 13] J. Pont-Tuset, F. Marqués. – Measures and meta-measures for the supervised evaluation of image segmentation. – Computer Vision and Pattern Recognition (CVPR), 06/2013 2013.

[Preetha 12] M.M.S.J. Preetha, L.P. Suresh, M.J. Bosco. – Image segmentation using seeded region growing. – Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on, pp. 576–583, 2012.

[Pressigout 06] Muriel Pressigout. – Approches hybrides pout le suivi temps-rêel d'objets complexes dans des séquences vidéo. – PhD. Thesis, Université de Rennes 1, 2006.

[Rangarajan 91] K. Rangarajan, M. Shah. – Establishing motion correspondence. – Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on, pp. 103–108, 1991.

[Roberts 63] Lawrence G Roberts. – Machine perception of three-dimensional solids. – Rapport de recherche, DTIC Document, 1963.

[Roerdink 00] Jos B.T.M. Roerdink, Arnold Meijster. – The watershed transform : Definitions, algorithms and parallelization strategies. Fundam. Inf., 41(1,2) :187–228, avril 2000.

[Rother 00] Carsten Rother. – A new approach for vanishing point detection in architectural environments. – In Proc. 11th British Machine Vision Conference, pp. 382–391, 2000.

[Rother 06] C. Rother, T. Minka, A. Blake, V. Kolmogorov. – Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. – Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 993–1000, 2006.

[Rubio 13] JoseC. Rubio, Joan Serrat, Antonio Lapez. – Video cosegmentation. Computer Vision - ACCV 2012, éd. par KyoungMu Lee, Yasuyuki Matsushita, JamesM. Rehg, Zhanyi Hu, pp. 13–24. – Springer Berlin Heidelberg, 2013.

[Salembier 95] Philippe Salembier, Jean Serra. – Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing, 4(8) :1153–1160, 1995.

[Schettini 93] Raimondo Schettini. – A segmentation algorithm for color images. Pattern Recognition Letters, 14(6) :499 – 506, 1993.

[Sekkal 12] Rafik Sekkal, Clément Strauss, François Pasteau, Marie Babel, Olivier Déforges. – Fast pseudo-semantic segmentation for joint region-based hierarchical and multiresolution representation. – Proc. of SPIE Electronic Imaging - Visual Communications and Image Processing, pp. 1–6, San Francisco, États-Unis, janvier 2012.

[Sekkal 13] R. Sekkal, F. Pasteau, M. Babel, B. Brun, I. Leplumey. – Simple monocular door detection and tracking. – IEEE Int. Conf. on Image Processing, ICIP'14, Melbourne, Australia, September 2013. IEEE.

[Sharon 06] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, Achi Brandt. – Hierarchy and adaptivity in segmenting visual scenes. Nature, 442(7104) :810–813, juin 2006.

[Shi 94a] J. Shi, C. Tomasi. – Good features to track. – Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pp. 593–600, 1994.

[Shi 94b] J. Shi, C. Tomasi. – Good features to track. – Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pp. 593–600, 1994.

[Shi 98] J. Shi, J. Malik. – Motion segmentation and tracking using normalized cuts. – Computer Vision, 1998. Sixth International Conference on, pp. 1154–1160, 1998.

[Shi 00a] J. Shi, J. Malik. – Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8) :888–905, 2000.

[Shi 00b] Jianbo Shi, Jitendra Malik. – Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8) :888–905, août 2000.

[Shi 06] Wenxia Shi, J. Samarabandu. – Investigating the performance of corridor and door detection algorithms in different environments. – Information and Automation, 2006. ICIA 2006. International Conference on, pp. 206–211, 2006.

[Shih 05] Frank Y. Shih, Shouxian Cheng. – Automatic seeded region growing for color image segmentation. Image and Vision Computing, 23(10) :877 – 886, 2005.

[Shineier 82] Shineier, Michael. – Extracting linear features from images using pyramids. Systems, Man and Cybernetics, IEEE Transactions on, 12(4) :569–572, 1982.

[Shotton 09] Jamie Shotton, John Winn, Carsten Rother, Antonio Criminisi. – Textonboost for image understanding : Multiclass object recognition and segmentation by jointly modeling texture, layout, and context. International Journal of Computer Vision, 81(1) :2–23, 2009.

[Shufelt 99] J.A. Shufelt. – Performance evaluation and analysis of vanishing point detection techniques. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(3) :282–288, 1999.

[Stoeter 00] S.A. Stoeter, F. Le Mauff, N.P. Papanikolopoulos. – Real-time door detection in cluttered environments. – Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on, pp. 187–192, 2000.

[Stojmenovic 10] M. Stojmenovic, A. Solis-Montero, A. Nayak. – Colour and texture based pyramidal image segmentation. – Audio Language and Image Processing (ICALIP), 2010 International Conference on, pp. 778–786, 2010.

[Strauss 10] C. Strauss, F. Pasteau, M. Babel, O. Déforges, L. Bédat. – Improved Image Partitioning for Compression and Re-

presentation using the Lab Color Space in the LAR Image Codec. – Proceedings of EUSIPCO 2010, pp. 1–5, Aalborg, Danemark, August 2010.

[Tang 10] Jun Tang. – A color image segmentation algorithm based on region growing. – Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, vol. 6, pp. V6–634–V6–637, 2010.

[Tanimoto 78] S.L. Tanimoto. – An optimal algorithm for computing fourier texture descriptors. Computers, IEEE Transactions on, C-27(1) :81–84, 1978.

[Teulière 10] C. Teulière. – Approches déterministes et bayésiennes pour un suivi robuste : application à l'asservissement visuel d'un drone. – PhD. Thesis, Université de Rennes 1, Mention traitement du signal et télécommunication, December 2010.

[Tian 10] Yingli Tian, Xiaodong Yang, Aries Arditi. – Computer vision-based door detection for accessibility of unfamiliar environments to blind persons. Computers Helping People with Special Needs, éd. par Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, Arthur Karshmer, pp. 263–270. – Springer Berlin Heidelberg, 2010.

[Tomasi 91] Carlo Tomasi, Takeo Kanade. – Detection and Tracking of Point Features. – Rapport de recherche, International Journal of Computer Vision, 1991.

[Tomita 73] Fumiaki Tomita, Masahiko Yachida, Saburo Tsuji. – Detection of homogeneous regions by structural analysis. – Proceedings of the 3rd international joint conference on Artificial intelligence, IJCAI'73, pp. 564–571, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc.

[Tu 05] Zhuowen Tu. – Probabilistic boosting-tree : learning discriminative models for classification, recognition, and clustering. – Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, pp. 1589–1596 Vol. 2, 2005.

[Ursani 08] Ahsan Ursani. – Fusion multiniveau pour la classification d'images de télédétection à très haute résolution spatiale. – PhD. Thesis, 2008, 1 vol.136 p.p. Thèsse de doctorat dirigée par Ronsin, Joseph et Kpalma, Kidiyo Traitement du signal et de l'image Rennes, INSA 2008.

[Vantaram 09a] S.R. Vantaram, E. Saber, S. Dianat, M. Shaw, R. Bhaskar. – An adaptive and progressive approach for efficient gradient-based multiresolution color image segmentation. – Image Processing (ICIP), 2009 16th IEEE International Conference on, pp. 2369–2372, 2009.

[Vantaram 09b] Sreenath Rao Vantaram, Eli Saber, Vincent Amuso, Mark Shaw, Ranjit Bhaskar. – Unsupervised image segmentation by automatic gradient thresholding for dynamic region growth in the cie l*a*b* color space. pp. 724019–724019–11, 2009.

[Ves 07] Esther Ves, Ana Ruedin, Daniel Acevedo, Xaro Benavent, Leticia Seijas. – A new wavelet-based texture descriptor for image retrieval. Computer Analysis of Images and Patterns, éd. par WalterG. Kropatsch, Martin Kampel, Allan Hanbury, pp. 895–902. – Springer Berlin Heidelberg, 2007.

[Vicente 10] Sara Vicente, Vladimir Kolmogorov, Carsten Rother. – Cosegmentation revisited : models and optimization. – Proceedings of the 11th European conference on Computer vision : Part II, ECCV'10, pp. 465–479, Berlin, Heidelberg, 2010. Springer-Verlag.

[Vilaplana 08] V. Vilaplana, F. Marques, P. Salembier. – Binary partition trees for object detection. Image Processing, IEEE Transactions on, 17(11) :2201–2216, 2008.

[Vincze 01] Markus Vincze. – Robust tracking of ellipses at frame rate. Pattern Recognition, 34(2) :487 – 498, 2001.

[vonGioi 12] R.G. von Gioi, J. Jakubowicz, J. M Morel, G. Randall. – LSD : a Line Segment Detector. Image Processing On Line, 2012, 2012.

[Wang 94] J.Y.A. Wang, E.H. Adelson. – Representing moving images with layers. Image Processing, IEEE Transactions on, 3(5) :625–638, 1994.

[Wang 98] Demin Wang. – Unsupervised video segmentation based on watersheds and temporal tracking. Circuits and Systems for Video Technology, IEEE Transactions on, 8(5) :539–546, 1998.

[Wirjadi 07] Oliver Wirjadi. – Survey of 3d image segmentation methods. – ITWM, 2007.

[Xiaohan 92] Yu Xiaohan, Juha Yla-Jaaski, O. Huttunen, T. Vehkomaki, O. Sipila, T. Katila. – Image segmentation combi-

ning region growing and edge detection. – Pattern Recognition, 1992. Vol.III. Conference C : Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on, pp. 481–484, 1992.

[Xu 12a] C. Xu, **J. J. Corso**. – Evaluation of super-voxel methods for early video processing. – Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[Xu 12b] C. Xu, C. Xiong, **J. J. Corso**. – Streaming hierarchical video segmentation. – Proceedings of European Conference on Computer Vision, 2012.

[Yang 10] Xiaodong Yang, YingLi Tian. – Robust door detection in unfamiliar environments by combining edge and corner features. – Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 57–64, 2010.

[Yilmaz 06] Alper Yilmaz, Omar Javed, Mubarak Shah. – Object tracking : A survey. ACM Comput. Surv., 38(4), décembre 2006.

[Yilmaz 08] A. Yilmaz, X. Li, M. Shah. – Object contour tracking using level sets, 2008.

[Yokoyama 05] M. Yokoyama, T. Poggio. – A contour-based moving object detection and tracking. – Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, pp. 271–276, 2005.

[Zhang 01] Dengsheng Zhang, Guojun Lu. – Segmentation of moving objects in image sequence : A review. – Circuits, Systems and Signal Process, pp. 143–183, 2001.

[Zhang 03] Zhihua Zhang, Chibiao Chen, Jian Sun, Kap Luk Chan. – {EM} algorithms for gaussian mixtures with split-and-merge operation. Pattern Recognition, 36(9) :1973 – 1983, 2003.

[Zhang 12] Jieyu Zhang, J.L. Barron. – Optical flow at occlusion. – Computer and Robot Vision (CRV), 2012 Ninth Conference on, pp. 198–205, 2012.

[Zucker 76] Steven W. Zucker. – Region growing : Childhood and adolescence. Computer Graphics and Image Processing, 5(3) :382 – 399, 1976.

Table des figures

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Techniques visuelles pour la détection et le suivi d'objets 2D

Nom Prénom de l'auteur : SEKKAL RAFIQ

Membres du jury :
- Monsieur FREMONT Vincent
- Monsieur RONSIN JOSEPH
- Madame PRESSIGOUT Muriel
- Monsieur MARQUES Ferran
- Monsieur BRUN Luc
- Madame BABEL Marie

Président du jury : RONSIN JOSEPH

Date de la soutenance : 28 Février 2014

Reproduction de la these soutenue

☑ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 28 Février 2014

Signature du président de jury

Le Directeur,

M'hamed DRISSI

Résumé

De nos jours, le traitement et l'analyse d'images trouvent leur application dans de nombreux domaines. Dans le cas de la navigation d'un robot mobile (fauteuil roulant) en milieu intérieur, l'extraction de repères visuels et leur suivi constituent une étape importante pour la réalisation de tâches robotiques (localisation, planification, etc.). En particulier, afin de réaliser une tâche de franchissement de portes, il est indispensable de détecter et suivre automatiquement toutes les portes qui existent dans l'environnement. La détection des portes n'est pas une tâche facile : la variation de l'état des portes (ouvertes ou fermées), leur apparence (de même couleur ou de couleur différentes des murs) et leur position par rapport à la caméra influe sur la robustesse du système. D'autre part, des tâches comme la détection des zones navigables ou l'évitement d'obstacles peuvent faire appel à des représentations enrichies par une sémantique adaptée afin d'interpréter le contenu de la scène. Pour cela, les techniques de segmentation permettent d'extraire des régions pseudo-sémantiques de l'image en fonction de plusieurs critères (couleur, gradient, texture…). En ajoutant la dimension temporelle, les régions sont alors suivies à travers des algorithmes de segmentation spatio-temporelle.

Dans cette thèse, des contributions répondant aux besoins cités sont présentées. Tout d'abord, une technique de détection et de suivi de portes dans un environnement de type couloir est proposée : basée sur des descripteurs géométriques dédiés, la solution offre de bons résultats. Ensuite, une technique originale de segmentation multirésolution et hiérarchique permet d'extraire une représentation en régions pseudosémantique. Enfin, cette technique est étendue pour les séquences vidéo afin de permettre le suivi des régions à travers le suivi de leurs contours. La qualité des résultats est démontrée et s'applique notamment au cas de vidéos de couloir.

Abstract

Nowadays, image processing remains a very important step in different fields of applications. In an indoor environment, for a navigation system related to a mobile robot (electrical wheelchair), visual information detection and tracking is crucial to perform robotic tasks (localization, planning…). In particular, when considering passing door task, it is essential to be able to detect and track automatically all the doors that belong to the environment. Door detection is not an obvious task: the variations related to the door status (open or closed), their appearance (e.g. same color as the walls) and their relative position to the camera have influence on the results. On the other hand, tasks such as the detection of navigable areas or obstacle avoidance may involve a dedicated semantic representation to interpret the content of the scene. Segmentation techniques are then used to extract pseudosemantic regions based on several criteria (color, gradient, texture...). When adding the temporal dimension, the regions are tracked then using spatiotemporal segmentation algorithms.

In this thesis, we first present joint door detection and tracking technique in a corridor environment: based on dedicated geometrical features, the proposed solution offers interesting results. Then, we present an original joint hierarchical and multiresolution segmentation framework able to extract a pseudo-semantic region representation. Finally, this technique is extended to video sequences to allow the tracking of regions along image sequences. Based on contour motion extraction, this solution has shown relevant results that can be successfully applied to corridor videos.

N° d'ordre : 14ISAR02/D14-02

Institut National des Sciences Appliquées de Rennes

20, Avenue des Buttes de Coësmes • CS 70839 • F-35708 Rennes Cedex 7

Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96