



**HAL**  
open science

# Exploration autonome et construction de cartes topologiques référencées vision omnidirectionnelle

Romain Marie

► **To cite this version:**

Romain Marie. Exploration autonome et construction de cartes topologiques référencées vision omnidirectionnelle. Sciences de l'ingénieur [physics]. Picardie Jules Verne University, 2014. Français. NNT: . tel-04515697

**HAL Id: tel-04515697**

**<https://hal.science/tel-04515697>**

Submitted on 21 Mar 2024

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

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

UNIVERSITÉ DE PICARDIE JULES VERNE  
ÉCOLE DOCTORALE SCIENCES ET SANTE

# T H È S E

pour obtenir le titre de

**Docteur en Sciences**

de l'Université de Picardie Jules Verne

**Mention : ROBOTIQUE**

Présentée et soutenue par

Romain MARIE

## Exploration autonome et construction de cartes topologiques référencées vision omnidirectionnelle

Thèse dirigée par Ouiddad LABBANI-IGBIDA et El Mustapha  
MOUADDIB

préparée au laboratoire Modélisation, Information et Systèmes

soutenue le 23 Avril 2014

### **Jury :**

<i>Président :</i>	Patrick RIVES	INRIA (Sophia Antipolis)
<i>Rapporteurs :</i>	Philippe MARTINET	IRCCYN (Nantes)
	Michel DEVY	LAAS-CNRS (Toulouse)
<i>Directeurs :</i>	Ouiddad LABBANI-IGBIDA	Université de Picardie Jules Verne
	El Mustapha MOUADDIB	Université de Picardie Jules Verne
<i>Examineur :</i>	Joaquim SALVI	University of Girona





## Remerciements

Avant d'entraîner le lecteur au coeur des préoccupations de la robotique mobile, je souhaite témoigner ma très sincère reconnaissance à l'ensemble des personnes qui ont contribué à leur façon à faire de cette thèse une expérience enrichissante, très formatrice, et surtout pleine de rencontres. Mes premiers mots vont sans l'ombre d'une hésitation à Ouiddad et Mustapha, qui ont réussi (je crois?) à me supporter tout ce temps. J'ai beaucoup appris ces dernières années, et je sais que c'est en quasi totalité grâce à vos conseils, votre science, et votre patience, que j'arrive aujourd'hui avec l'étrange sentiment d'avoir compris ce sur quoi j'ai passé tant de temps.

Je remercie bien sûr également la Direction Générale de l'Armement, ainsi que le Fond Européen de Développement Régional en Picardie, qui m'ont accordé leur confiance, et, à travers le financement qu'ils m'ont accordé, m'ont permis de dédier mon temps et mon énergie à la science.

Mes pensées vont ensuite à l'équipe des doctorants PR, auquel je rajoute bien sûr Cyril, Guillaume et Florian, qui, entre Omnibeer et pauses cafés, ont apporté juste la bonne dose de bonne humeur à des journées parfois un peu difficiles. Je remercie bien sûr le laboratoire MIS dans son ensemble, pour m'y avoir fait toujours sentir à mon aise, et Valérie en particulier, pour être toujours si disponible, réactive et efficace.

Je tiens également à exprimer mes sincères remerciements à l'ensemble de l'équipe mécatronique du laboratoire XLIM, qui, m'a accueilli à bras ouverts. Si je me sens si bien dans cette nouvelle ville, c'est en partie grâce à vous.

Dans un registre plus personnel, je remercie mes parents pour leur soutien qui date de bien avant la thèse, et qui n'a jamais fléchi. Je garde bien entendu les dernières lignes pour Marie-Hélène, qui assume à merveille la lourde charge de me supporter. Je n'y serais pas arrivé sans toi. Ce mémoire est donc un peu le tien !



# Table des matières

<b>1</b>	<b>Introduction générale et mise en contexte des travaux</b>	<b>1</b>
1.1	Présentation du contexte . . . . .	1
1.2	Positionnement du travail . . . . .	2
1.2.1	Analyse de la problématique . . . . .	2
1.2.2	Hypothèses et enjeux . . . . .	2
1.3	Principales contributions . . . . .	4
1.4	Organisation du mémoire . . . . .	5
<b>2</b>	<b>Vision omnidirectionnelle : Modèle de projection et extraction de l'espace libre</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	La vision omnidirectionnelle . . . . .	8
2.2.1	Définition . . . . .	8
2.2.2	Dispositifs existants . . . . .	8
2.3	Capteurs catadioptriques : modélisation . . . . .	11
2.3.1	La contrainte du point de vue unique . . . . .	11
2.3.2	Modèle de projection unifié . . . . .	11
2.3.3	Vers une image digitale . . . . .	13
2.4	Extraction de l'espace libre . . . . .	14
2.4.1	Espace libre omnidirectionnel : Définition . . . . .	14
2.4.2	Approche par contours actifs . . . . .	14
2.4.3	Vue d'ensemble de l'algorithme . . . . .	15
2.4.4	Résultats . . . . .	17
2.4.5	Discussion . . . . .	17
2.5	Conclusion . . . . .	17
<b>3</b>	<b>Topologie locale par calcul d'axe médian</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	L'axe médian . . . . .	22
3.2.1	Première approche et formalisations existantes . . . . .	22
3.2.2	Propriétés fondamentales . . . . .	24
3.2.3	Axe médian, squelette et transformée en axe médian . . . . .	25
3.3	Etat de l'art . . . . .	26
3.3.1	Extraction d'axe médian . . . . .	26
3.3.2	Filtrage des branches parasites . . . . .	28
3.3.3	Cadres applicatifs . . . . .	29
3.3.4	Discussion . . . . .	30
3.4	Le Delta Medial axis . . . . .	30
3.4.1	Définition des objectifs . . . . .	30
3.4.2	Formulation . . . . .	31

3.4.3	Implémentation et optimisations . . . . .	32
3.4.4	Résultats expérimentaux . . . . .	38
3.4.5	Première application : cartographie topologique d'une vue aérienne . . . . .	43
3.5	Adaptation aux images omnidirectionnelles . . . . .	46
3.5.1	Distortions et axe médian . . . . .	46
3.5.2	Projection inverse . . . . .	48
3.5.3	Vers une métrique adaptée . . . . .	49
3.5.4	Formalisation du ODMA . . . . .	49
3.5.5	Réglage automatique de $\tilde{\delta}$ . . . . .	50
3.5.6	Quid de l'implémentation ? . . . . .	51
3.5.7	Résultats . . . . .	53
3.6	Conclusion . . . . .	55
<b>4</b>	<b>Signature visuelle par invariant intégral</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	Indexation de lieu par construction de signature . . . . .	60
4.2.1	La notion de lieu . . . . .	60
4.2.2	Invariance, discriminance et complétude . . . . .	62
4.2.3	Etat de l'art . . . . .	62
4.3	Invariants intégraux et signature visuelle . . . . .	65
4.3.1	Principe général et formalisme de Haar . . . . .	65
4.3.2	Utilisations existantes . . . . .	65
4.4	Elaboration de la signature . . . . .	66
4.4.1	Introduction . . . . .	66
4.4.2	La configuration $\mathbf{x}$ . . . . .	67
4.4.3	Le groupe de transformation $\mathcal{G}$ . . . . .	68
4.4.4	Le noyau $f$ . . . . .	71
4.4.5	Expression finale de la signature . . . . .	78
4.5	Calcul effectif, implémentation et optimisations . . . . .	79
4.5.1	Analyse préalable : les différentes étapes . . . . .	79
4.5.2	Calculs réalisés hors ligne . . . . .	79
4.5.3	Optimisation des calculs en ligne . . . . .	80
4.6	Mesure de distance entre signatures . . . . .	81
4.7	Validation expérimentale . . . . .	83
4.7.1	Avant-propos . . . . .	83
4.7.2	Scénario en intérieur . . . . .	83
4.7.3	Scénario en extérieur . . . . .	89
4.8	Conclusion . . . . .	93
<b>5</b>	<b>Navigation, exploration et représentation de l'environnement</b>	<b>95</b>
5.1	Introduction . . . . .	95
5.2	Représentation de l'environnement . . . . .	96
5.2.1	Les solutions existantes . . . . .	96

---

5.2.2	Notre représentation spatiale topologique et photométrique . . .	97
5.3	Processus incrémental de construction de la carte . . . . .	100
5.3.1	Analyse de la problématique . . . . .	100
5.3.2	(1) : Le départ d'un noeud topologique . . . . .	102
5.3.3	(2) : La découverte d'un nouveau lieu . . . . .	103
5.3.4	(3) et (4) : L'arrivée à un noeud topologique . . . . .	104
5.4	Localisation dans la carte . . . . .	110
5.4.1	Localisation topologique du robot . . . . .	110
5.4.2	Localisation photométrique et gestion des fermetures de boucle	110
5.4.3	Identification de l'orientation des branches du squelette . . . .	113
5.5	Déplacements dans l'environnement . . . . .	113
5.6	Exemple d'application : exploration et couverture de l'espace . . . .	115
5.7	Conclusion . . . . .	117
<b>6</b>	<b>Conclusion générale et perspectives</b>	<b>121</b>
6.1	Bilan général . . . . .	121
6.2	Perspectives . . . . .	122
6.3	Valorisation des travaux de thèse . . . . .	123
	<b>Bibliographie</b>	<b>125</b>



# Introduction générale et mise en contexte des travaux

---

## Sommaire

---

<b>1.1</b>	<b>Présentation du contexte</b> . . . . .	<b>1</b>
<b>1.2</b>	<b>Positionnement du travail</b> . . . . .	<b>2</b>
1.2.1	Analyse de la problématique . . . . .	2
1.2.2	Hypothèses et enjeux . . . . .	2
<b>1.3</b>	<b>Principales contributions</b> . . . . .	<b>4</b>
<b>1.4</b>	<b>Organisation du mémoire</b> . . . . .	<b>5</b>

---

## 1.1 Présentation du contexte

Les travaux présentés dans ce mémoire ont été réalisés au sein du laboratoire Modélisation, Information et Systèmes (MIS) de l'Université de Picardie Jules Verne grâce à un cofinancement de la Direction Générale de l'Armement (DGA) et du Fond Européen de Développement Régional (FEDER) en Picardie<sup>1</sup>. Bien qu'il soit parvenu à son terme l'an passé, ces travaux s'intègrent directement dans les thématiques de recherche portées par le projet ANR R-Discover (Mobile **R**obot network : **D**ecentralized omnivision-based space **c**overage. Cooperative perception, localization and navigation). Ce projet d'envergure réunissait quatre laboratoires (MIS, Lasma, Greyc et LIP6) et un groupe industriel (Thalès), et avait pour objectif, à partir d'un drone et d'un ensemble de robots mobiles terrestres, l'exploration et la couverture visuelle d'un environnement initialement inconnu, avec pour principale modalité de perception (pour les robots au sol) un capteur de vision omnidirectionnelle.

Une grande partie du travail réalisé au cours de ces trois années porte en particulier sur l'utilisation dudit capteur, et vise à fournir aux robots terrestres un ensemble de représentations qui leur offre une complète autonomie dans la réalisation de leur mission.

---

1. Nous remercions chaleureusement ces deux organismes pour le financement qu'ils nous ont accordé.



## 1.2 Positionnement du travail

### 1.2.1 Analyse de la problématique

Pour être considéré comme autonome, un robot mobile doit être capable de se déplacer de manière sûre, sans aucune intervention humaine. A partir de sa position courante, il doit pouvoir définir, et emprunter, une trajectoire le menant à n'importe quelle position connue de son environnement. Ce processus porte le nom de navigation, et se résume à la résolution de trois questions [Leonard 1991] : *Où suis-je ?*, *Où vais-je ?*, *Comment puis-je m'y rendre ?*.

Ces questions font échos à quatre tâches fondamentales en robotique mobile, ayant chacune donné lieu à une littérature très fournie. Il s'agit de :

- **la localisation**, qui vise à estimer la position du robot relativement à l'environnement dans lequel il se trouve ;
- **la cartographie**, qui consiste à élaborer un modèle interne de l'environnement compréhensible du robot ;
- **la planification de trajectoire**, qui définit à partir de la carte construite les déplacements à effectuer pour atteindre une position destination ;
- **la commande**, qui tente de faire coïncider les mouvements du robot avec la trajectoire prévue dans la représentation.

Au cours de cette thèse, nous avons travaillé sur chacun de ces points. Nous développons dans ce mémoire une solution complète au problème de la navigation autonome, étroitement liée au contexte de la mission. Si la littérature est riche sur le sujet, c'est en effet que l'élaboration d'une réponse aux trois questions évoquées plus tôt peut varier du tout au tout en fonction 1) des spécificités de l'environnement, 2) des choix matériels et logiciels relatifs au robot, et 3) des caractéristiques de la mission qui lui est attribuée.

### 1.2.2 Hypothèses et enjeux

Le premier point à définir est la connaissance du robot sur son environnement. Dans [DeSouza 2002], les auteurs classifient les méthodes de navigation suivant trois degrés de complexité :

- les approches sans carte, qui ne permettent qu'un comportement réactif du robot ;
- les approches avec carte, qui considèrent le modèle interne de l'environnement comme au moins partiellement acquis a priori, ce qui simplifie grandement la problématique de la cartographie ;
- les approches de construction incrémentale de la représentation, qui nécessitent de définir à chaque instant si la position courante du robot est connue ou non.

Notre travail s'inscrit dans ce dernier niveau de complexité, puisque, d'une part, le robot ne possède à l'état initial aucun modèle de l'environnement, et, d'autre

part, une navigation réactive n'est pas considérée comme suffisante. Il va donc lui falloir construire, en temps réel, une représentation de l'environnement qui indexe les informations qu'il perçoit à chaque instant.

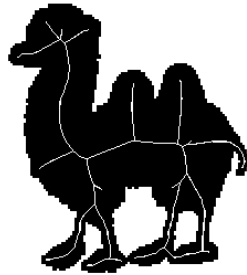
Se pose ensuite la question des modalités de perception. Pour mener à bien sa mission, un robot a besoin d'acquérir des informations qui lui permettent de comprendre l'environnement dans lequel il évolue. Par exemple, pour se localiser, il doit définir une empreinte de la scène qu'il observe, afin de la comparer avec celle des lieux qu'il a déjà eus l'occasion de visiter. De même, pour se déplacer sans risque dans l'environnement, il a besoin d'identifier la position des différents obstacles qui l'entourent. A partir de certains capteurs, en particulier les systèmes actifs comme les télémètres laser, la question de la navigation est considérée par la communauté comme pratiquement résolue (au moins en environnement structuré de dimensions limitées). Pour d'autres modalités, notamment les capteurs de vision, il s'agit en revanche d'une problématique ouverte, qui donne lieu à une littérature très abondante. Cette modalité fournit en effet une information très riche sur la scène perçue, mais qui se révèle dans la pratique difficile à extraire. Dans ce travail, nous utilisons un capteur de vision catadioptrique, qui, au prix de distortions importantes dans l'image résultante, fournit une perception omnidirectionnelle de l'environnement observé. L'un des enjeux majeurs de cette thèse est, à partir d'un capteur catadioptrique, d'extraire à chaque instant l'ensemble des informations nécessaires à la résolution des problématiques évoquées plus tôt.

Enfin, le dernier point à considérer repose sur les spécificités de la mission liées à son contexte. Bien que cette notion ne soit pas explicitement utilisée, nous avons toujours conservé à l'esprit l'extension du travail à un contexte multi-robots, en particulier dans l'élaboration des représentations spatiales. Elles sont ainsi ancrées dans l'environnement, et définies de façon à faciliter leur partage entre plusieurs robots. De même, la contrainte du temps réel a toujours été au coeur des réflexions, et a ainsi mené l'ensemble de nos choix vers des solutions algorithmiques à la complexité linéaire (ou quasi-linéaire).

### 1.3 Principales contributions

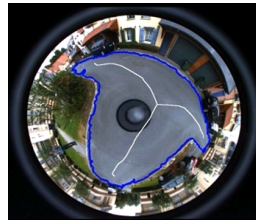
Le travail présenté dans ce mémoire repose sur l'utilisation d'un capteur de vision omnidirectionnelle, qui définit l'ensemble des entrées pour un processus complet d'exploration et de cartographie d'un environnement inconnu a priori. Les principales contributions de cette thèse sont :

- **Le Delta Medial Axis :**



Né du besoin d'estimer en temps réel le squelette d'une forme 2d définie dans un espace discret (l'image et susceptible d'être affectée par des déformations (bruit d'acquisition ou de segmentation), cet outil dépasse de loin le cadre de la robotique mobile. Il intègre un processus original d'élagage fondé sur un seul paramètre qui, comme nous le verrons, présente une stabilité face aux déformations significativement supérieure à celle des méthodes de l'état de l'art de complexité comparable.

- **L'extraction dans l'image du squelette de l'espace navigable proximal :**



Cette première représentation est fondée sur l'étude de l'espace navigable perçu. L'idée est d'y appliquer une version adaptée du *Delta Medial Axis* qui prend en compte la géométrie particulière du capteur afin d'extraire dans l'image le diagramme de Voronoï de la scène observée. De cette façon, le robot capture à chaque instant la topologie de la scène qu'il observe, et obtient un ensemble de trajectoires qui maximisent la distance aux obstacles.

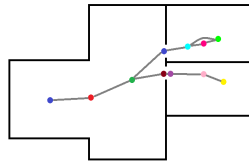
- **L'élaboration d'une signature visuelle adaptée :**



Cette seconde représentation vise à capturer l'information photométrique perçue, afin d'en construire une signature qui sert d'index dans un processus de

localisation. Elle s'appuie sur le formalisme des invariants intégraux de Haar, et sélectionne l'information visuelle la plus riche en ignorant la portion de l'image qui représente l'espace libre (souvent homogène et commune à différents lieux). En considérant uniquement son complément (obstacles et objets visibles), la signature construite capture l'information pertinente, et réduit ainsi le risque d'ambiguïté perceptive. De plus, les conditions particulières d'acquisition sont intégrées au processus de construction en pondérant l'impact de chaque point par sa position dans l'image et sa distance au robot.

- **La représentation topologique de l'environnement :**



Les deux représentations sont utilisées conjointement pour former une carte topologique de l'environnement qui sert à la prise de décision dans un processus de couverture de l'espace. Les noeuds sont définis à partir des signatures, et modélisent les différents lieux de l'environnement. Les arêtes sont induites par les déplacements sur le squelette de l'espace libre, et établissent les relations d'adjacence qui lient les noeuds. Cette connaissance globale offre au robot la possibilité de ne pas se contenter d'une navigation réactive, en ajoutant des possibilités décisionnelles à plus grande échelle qui lui permettent par exemple de choisir une direction lorsqu'il atteint une intersection.

## 1.4 Organisation du mémoire

Ces différentes contributions ayant toutes été élaborées en visant le même objectif, il est logique qu'elles ne soient pas indépendantes. La figure 1.1 présente les différentes tâches qui les lient les unes aux autres.

Le mémoire est alors structuré comme suit :

**Le chapitre 2** porte sur la vision omnidirectionnelle en général, et sur les capteurs catadioptriques en particulier. Outre l'analyse du processus de projection de la scène à travers ce type de dispositif, le discours s'arrête sur un algorithme développé au sein du laboratoire, et qui permet, à partir de l'image omnidirectionnelle, d'extraire l'espace navigable proximal du robot. Cet algorithme est la clé de voûte de presque tous les outils développés au cours de cette thèse, et nécessite d'être introduit.

**Le chapitre 3** s'intéresse à la première des représentations locales construites à chaque instant par le robot. Il présente ainsi le cheminement menant à l'extraction du squelette de l'espace navigable directement dans l'image, en partant de la définition du Delta Medial Axis, jusqu'à son adaptation aux images omnidirectionnelles. Il développe également une seconde application du DMA directement liée au

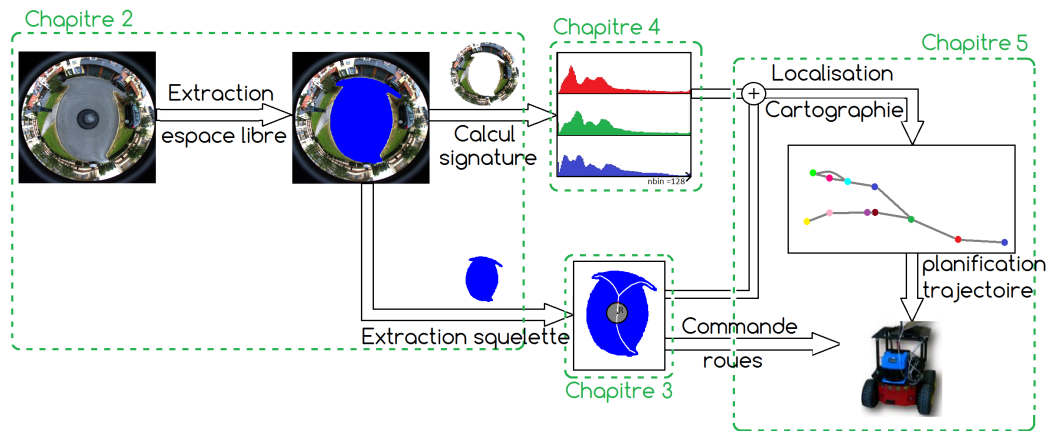


FIGURE 1.1 – Organisation générale des représentations et illustration des connexions entre les différents chapitres.

contexte du travail, à travers l'élaboration d'une carte topologique d'un environnement, depuis une vue aérienne de ce dernier.

**Le chapitre 4** propose de s'intéresser au problème de la construction de signature visuelle. Cette question très vaste fera l'objet d'un état de l'art, en particulier porté sur l'opposition entre les approches locales et globales. Une solution sera ensuite proposée en utilisant le formalisme des invariants intégraux de Haar. Ce dernier impliquant un ensemble de choix, chacun fera l'objet d'une analyse approfondie.

**Le chapitre 5** présente la solution retenue pour permettre au robot d'avoir un modèle interne de l'environnement qu'il a déjà exploré. Nous verrons en particulier comment les représentations locales s'articulent sous la forme d'une carte topologique où le robot est capable de définir des itinéraires, et de les transformer en trajectoires dans un processus de couverture de l'espace.

**Le chapitre 6** fera enfin office de bilan et de conclusion. Nous reviendrons sur l'ensemble du travail réalisé, et suggérerons en perspective de nombreuses pistes permettant de le poursuivre.

# Vision omnidirectionnelle :

## Modèle de projection et

## extraction de l'espace libre

---

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>7</b>
<b>2.2</b>	<b>La vision omnidirectionnelle</b>	<b>8</b>
2.2.1	Définition	8
2.2.2	Dispositifs existants	8
<b>2.3</b>	<b>Capteurs catadioptriques : modélisation</b>	<b>11</b>
2.3.1	La contrainte du point de vue unique	11
2.3.2	Modèle de projection unifié	11
2.3.3	Vers une image digitale	13
<b>2.4</b>	<b>Extraction de l'espace libre</b>	<b>14</b>
2.4.1	Espace libre omnidirectionnel : Définition	14
2.4.2	Approche par contours actifs	14
2.4.3	Vue d'ensemble de l'algorithme	15
2.4.4	Résultats	17
2.4.5	Discussion	17
<b>2.5</b>	<b>Conclusion</b>	<b>17</b>

---

## 2.1 Introduction

Les informations que possède un robot sur l'environnement dans lequel il évolue sont en général la clé d'une bonne réalisation de sa mission. Elles peuvent provenir de deux origines différentes :

- une connaissance a priori du scénario, comme par exemple une carte de l'environnement qui lui serait fournie par l'utilisateur
- un ensemble de données acquises en temps réel par ses modalités de perception.

Dans notre cas, la connaissance a priori du scénario est nulle. Le robot doit donc se contenter des informations qu'il acquiert par lui-même. Ce chapitre est consacré à l'étude de la vision omnidirectionnelle en général, et des capteurs catadioptriques en

particulier, puisque la seule capacité de perception du robot provient de l'un d'entre eux. Nous nous attacherons en particulier à comprendre le processus de formation de l'image à travers le modèle de projection unifié [Barreto 2001][Geyer 2001]. La seconde partie du chapitre présente un algorithme d'extraction de l'espace libre directement dans l'image omnidirectionnelle, extrait de la thèse de Pauline Merveilleux [Merveilleux 2013] soutenue en 2012. Il représente la clé de voûte des contributions majeures de ce travail, et nécessite d'être présenté.

## 2.2 La vision omnidirectionnelle

### 2.2.1 Définition

Un système de vision est considéré comme omnidirectionnel si son champ d'observation couvre l'ensemble d'une sphère dont il est le centre. Il doit pour cela être capable de fournir une indication photométrique concernant chacune des directions au départ du capteur. Dans la pratique, on fait souvent l'amalgame entre vision omnidirectionnelle et vision panoramique [Mouaddib 2005], qui désigne tout dispositif dont le champ de vue couvre  $360^\circ$  autour de l'axe vertical. La figure 2.1 présente un exemple de panorama construit à partir du capteur utilisé dans ce travail.



FIGURE 2.1 – Exemple de vue panoramique obtenue dans le cadre des travaux de thèse.

### 2.2.2 Dispositifs existants

Différents dispositifs permettent de générer un tel champ d'observation.

#### 2.2.2.1 Les systèmes motorisés

Une première approche consiste à fixer une caméra perspective à un dispositif motorisé qui permet d'orienter le champ de vue à volonté. L'idée est alors de réaliser l'acquisition d'un ensemble de prises de vues, qui, mises bout à bout, assurent l'omnidirectionnalité. Par exemple, dans [Romain 2001] et [Ea 2001] (voir Fig 2.2), deux systèmes de stéréovision, l'un panoramique et l'autre omnidirectionnel (les auteurs parlent de "sphéro-vision") sont présentés. Le premier utilise un axe de rotation autour de l'axe vertical, tandis que le second en couple deux. Ce type de dispositif est néanmoins encombrant, ce qui le rend peu adapté à des applications en robotique mobile, d'autant plus que la phase d'acquisition est beaucoup trop lente pour

envisager la moindre application temps réel. De plus, la fusion des différentes prises de vue n'est pas un problème trivial, puisque même en supposant le dispositif parfaitement immobile, l'environnement peut subir des modifications pendant la phase d'acquisition (les conditions d'illumination peuvent varier, ou des objets peuvent être en mouvement dans la scène).

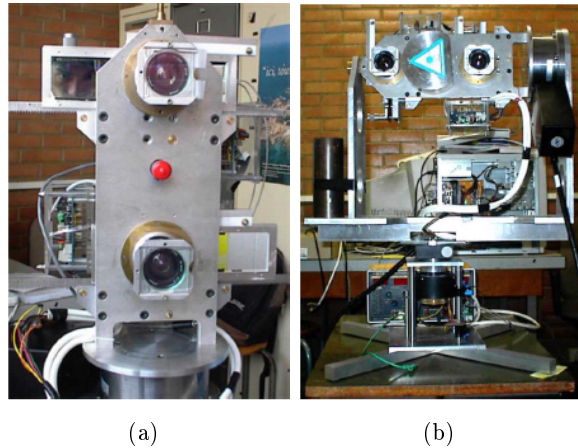


FIGURE 2.2 – Exemples de capteurs omnidirectionnels motorisés. (a) système de stéréovision panoramique [Romain 2001]. (b) système de stéréovision omnidirectionnelle [Ea 2001].

### 2.2.2.2 Les capteurs polydioptriques

Une alternative permettant de résoudre les problèmes liés au décalage temporel entre les prises de vue consiste à utiliser conjointement un réseau de caméras fixes. Placées afin de couvrir l'ensemble du champ d'observation, elles forment un dispositif rigide qui facilite le processus de construction de la vue omnidirectionnelle. Un exemple actuel est développé par la société Google. Dans sa volonté de photographier la planète, elle utilise en effet un dispositif poly-dioptrique qu'elle installe sur différents moyens de locomotion, le plus connu étant sans conteste la voiture Google. Ce type de système est néanmoins à la fois encombrant et onéreux, ce qui le rend peu adapté à notre contexte.

### 2.2.2.3 Les optiques oeil de poisson

Il y a ensuite les optiques "oeil de poisson" (*fisheye* en anglais). Appelées également objectifs hypergone ou grand angle, ils ont pour particularité d'avoir une distance focale très courte, qui induit un angle de champ très important. Le principal avantage d'un tel système est sa simplicité d'utilisation, puisqu'une image issue d'une seule caméra génère la vue panoramique convoitée. En contrepartie, ce type de capteur ne vérifie pas la contrainte du point de vue unique (dont l'importance sera discutée dans la section suivante), et le champ de vue est en général inférieur



aux autres solutions existantes. Un objectif de ce type ainsi que l'image résultante sont présentés Fig.2.3.

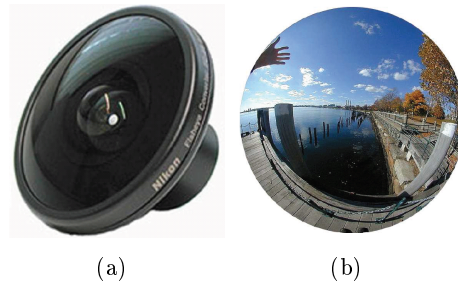


FIGURE 2.3 – a) exemple d'optique hypergone (Nikon-FC-E8) b) Image obtenue à partir de ce type d'objectif

#### 2.2.2.4 Les capteurs catadioptriques

Il y a enfin la famille des capteurs catadioptriques. L'idée est de coupler une caméra avec un miroir de révolution, afin d'observer la projection de la scène sur ce dernier. Un tel dispositif est à la fois compact (voif Fig.2.4(a)) et efficace puisque la scène entière est capturée en une prise de vue. Les images obtenues sont néanmoins sujettes à des distorsions importantes (un exemple est proposé Fig.2.4(b)) et nécessitent des traitements adaptés afin d'être correctement analysées.

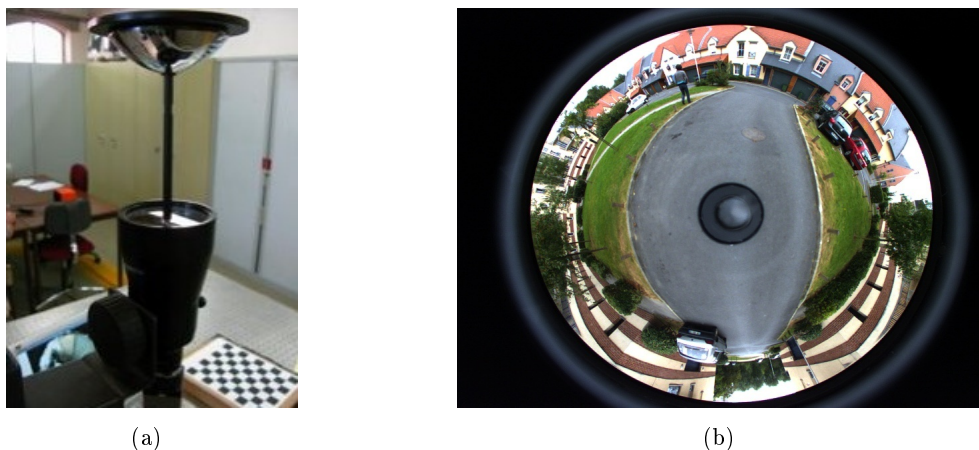


FIGURE 2.4 – Exemple (a) de capteur catadioptrique, et (b) d'image qu'il permet d'obtenir.

## 2.3 Capteurs catadioptriques : modélisation

### 2.3.1 La contrainte du point de vue unique

La contrainte du point de vue unique implique que les rayons lumineux qui convergent vers le plan image passent tous par un même point appelé centre de projection. Lorsqu'elle est vérifiée, chaque point du plan image correspond à un unique rayon lumineux, ce qui rend la modélisation du processus de projection beaucoup plus simple. Dans le cas des capteurs catadioptriques, cette contrainte n'est en rien acquise. L'image est en effet le résultat d'une double projection, d'abord sur le miroir, puis dans le plan image. Il existe néanmoins plusieurs configurations qui permettent de la vérifier. Parmi elles (présentées dans [Baker 1999]), deux sont de loin les plus populaires. La première (Figure 2.4(a)) couple un miroir parabololoïde à un objectif télécentrique, ce qui rend le dispositif relativement encombrant. La seconde (Figure 2.4(b)) associe un miroir hyperboloïde avec une caméra perspective, générant un dispositif à la fois plus compact et moins onéreux. Nous utilisons dans ce travail un capteur de ce type.

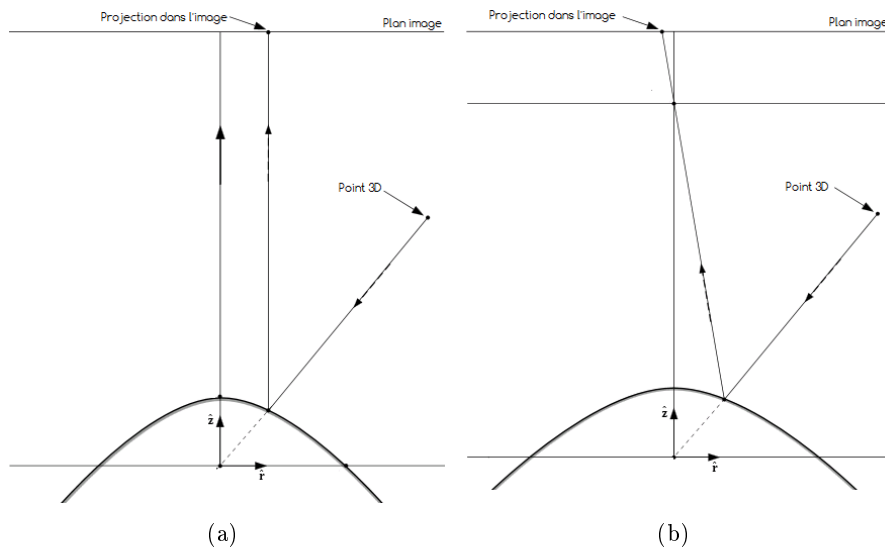


FIGURE 2.5 – Schématisation du processus de projection d'un point 3D vers le plan image, dans le cas (a) d'un dispositif caméra orthographique / miroir parabololoïde, et (b) d'un dispositif caméra perspective / miroir hyperboloïde.

### 2.3.2 Modèle de projection unifié

Les chapitres suivants nécessitent de comprendre le processus de formation de l'image, donc de modéliser la projection des points de la scène à travers le capteur. Dans [Barreto 2001], Barreto et Araujo développent un modèle de projection adapté à l'ensemble des systèmes de caméra à point de vue unique. Il consiste en deux projections successives, d'abord sur une sphère unitaire, puis dans le plan image, régies

par un ensemble d'équations que cette section regroupe afin d'alléger les développements à venir.

Considérons un point 3D de la scène  $\mathbf{P} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})^t$  défini dans le repère caméra. Sa projection sur la sphère unitaire notée  $\mathbf{P}_S = (\mathbf{X}_S, \mathbf{Y}_S, \mathbf{Z}_S)^t$  est représentée par la fonction  $f_{sph}$  qui s'écrit :

$$f_{sph} : \mathbf{P} \in \mathbb{R}^3 \mapsto \mathbf{P}_S \in \mathbb{S}^2, \quad (2.1)$$

$$\mathbf{P}_S = f_{sph}(\mathbf{P}) \text{ tel que } \begin{cases} \mathbf{X}_S = \frac{\mathbf{X}}{\|\mathbf{P}\|} \\ \mathbf{Y}_S = \frac{\mathbf{Y}}{\|\mathbf{P}\|} \\ \mathbf{Z}_S = \frac{\mathbf{Z}}{\|\mathbf{P}\|} \end{cases}$$

Où  $\|\mathbf{P}\|$  désigne la norme du point  $\mathbf{P}$  qui s'exprime par :  $\|\mathbf{P}\| = \sqrt{\mathbf{X}^2 + \mathbf{Y}^2 + \mathbf{Z}^2}$ .

La seconde étape de ce modèle projette  $\mathbf{P}_S$  dans le plan image normalisé en  $\mathbf{p} = (\mathbf{x}, \mathbf{y})^t$  par l'intermédiaire d'une projection perspective notée  $f_{img}$  centrée en  $C = (0, 0, \xi)^t$  qui s'écrit :

$$f_{img} : \mathbf{P}_S \in \mathbb{S}^2 \mapsto \mathbf{p} \in \mathbb{R}^2, \quad (2.2)$$

$$\mathbf{p} = f_{img}(\mathbf{P}_S), \text{ avec } \begin{cases} x = X_s / (Z_s + \xi) \\ y = Y_s / (Z_s + \xi) \end{cases}$$

Le paramètre  $\xi$  dépend intrinsèquement du type de miroir utilisé (il vaudra par exemple 0 dans le cas d'un miroir plan et 1 pour un miroir parabolique) et peut être déterminé lors d'une phase de calibrage.

La composition des deux fonctions de projection  $f_{sph}$  et  $f_{img}$  traduit le trajet complet d'un point de la scène jusqu'au plan image normalisé. La fonction de projection  $f$  qui le modélise s'écrit :

$$f = f_{img} \circ f_{sph} : \mathbf{P} \in \mathbb{R}^3 \mapsto \mathbf{p} \in \mathbb{R}^2; t.q. \begin{cases} x = X / (Z + \|\mathbf{P}\|\xi) \\ y = Y / (Z + \|\mathbf{P}\|\xi) \end{cases} \quad (2.3)$$

Par défaut,  $f$  n'est pas bijective, puisqu'il est impossible à partir de la position dans l'image d'un point, de retrouver le point 3D correspondant (à moins de posséder des informations sur la structure de la scène observée). En revanche, et ce sera utilisé dans la suite du mémoire, la seconde projection  $f_{img}$  peut être inversée. Etant donné un point  $\mathbf{p}$  du plan image normalisé, les coordonnées du point  $\mathbf{P}_S$  correspondant sur la sphère d'équivalence sont données par :

$$\mathbf{P}_S = f_{img}^{-1}(\mathbf{p}) : \begin{cases} X_s = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} \cdot x \\ Y_s = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} \cdot y \\ Z_s = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} - \xi \end{cases} \quad (2.4)$$

La figure 2.6 propose une vue d'ensemble du modèle de projection utilisé.

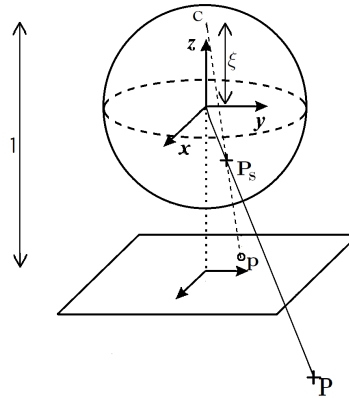


FIGURE 2.6 – Représentation schématisée du modèle de projection unifié [Barreto 2001]. Afin de rendre la figure plus lisible, les proportions ne sont pas respectées.

### 2.3.3 Vers une image digitale

Le point  $\mathbf{p}$  est défini dans un repère métrique dont l'origine est le centre de projection de la caméra. Mais d'un point de vue informatique, une image est une grille définie dans un espace discret, qui est la plupart du temps mémorisée sous la forme d'une matrice. La coordonnée  $(0, 0)$  correspond le plus souvent au point situé dans le coin supérieur gauche de l'image. La modélisation traduisant la projection de la scène en une image digitale est donc complétée par une dernière fonction qui traduit la transformation entre l'espace métrique et l'espace pixelique. On qualifie d'intrinsèques l'ensemble des paramètres qui permettent de réaliser cette transformation. Ils forment une matrice  $\mathbf{K}$ , nommée matrice des paramètres intrinsèques qui s'écrit :

$$\mathbf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

Les paramètres  $\alpha_u$  et  $\alpha_v$  modélisent le changement d'échelle qui s'opère entre les deux repères, et incorporent d'une part la focale  $f$  de la caméra, et de l'autre les dimensions (métriques) d'un pixel  $k_u$  et  $k_v$  (on a :  $\alpha_i = fk_i$ ,  $i = u, v$ ).  $u_0$  et  $v_0$  correspondent quant à eux aux coordonnées pixeliques du centre de projection de la caméra. Alors, l'obtention des coordonnées pixeliques  $(u, v, 1)^t$  du point  $\mathbf{p}$  est définie par :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{1} \end{pmatrix} \quad (2.6)$$

La matrice  $\mathbf{K}$ , ainsi que le paramètre  $\xi$  nécessitent d'être déterminés en amont pour permettre une modélisation exploitable du processus de projection. Une phase

de calibrage est donc nécessaire. Nous utilisons dans ce travail la solution logicielle HYSCAS [Caron 2011], développée au sein du laboratoire.

## 2.4 Extraction de l'espace libre

Les représentations construites dans les chapitres à venir sont toutes fondées sur l'utilisation d'une information de haut niveau extraite dans l'image : l'espace libre omnidirectionnel. Cette section décrit une approche algorithmique capable de l'extraire en temps réel [Merveilleux 2013].

### 2.4.1 Espace libre omnidirectionnel : Définition

Dans un contexte général, l'espace libre peut être vu comme la portion de l'environnement dans laquelle le robot peut se déplacer sans risque de collision avec un obstacle (région représentée en vert sur la figure 2.7(a)). Mais lorsque c'est au robot de l'extraire, cette définition nécessite d'être adaptée aux modalités de perception à sa disposition. Nous définissons donc ici l'espace libre tel que perçu dans l'image omnidirectionnelle, comme la région de l'image homogène au sens photométrique, qui englobe la projection du robot (en bleu Figure.2.7(b)).

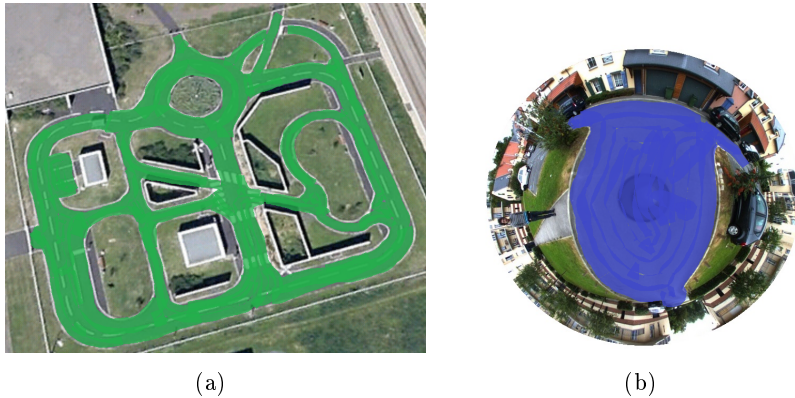


FIGURE 2.7 – a) Espace libre au sens de l'environnement (b) Espace libre au sens de l'image omnidirectionnelle

### 2.4.2 Approche par contours actifs

Au cours de ses travaux de thèse [Merveilleux 2010, Merveilleux 2011a, Merveilleux 2011b], Pauline Merveilleux a proposé et étudié trois modèles de contours actifs qui permettent l'extraction de l'espace libre directement dans l'image omnidirectionnelle :

- Un modèle paramétrique amélioré (qui intègre un processus de mise à jour de la densité des points appartenant au contour) réputé pour être rapide d'un

point de vue computationnel, mais incapable d'effectuer des segmentations multiples à partir d'un contour déformable

- Un modèle géométrique (level set) qui tolère les segmentations multiples et la gestion des régions à très forte courbure, mais le paie par un temps de calcul très important.
- Un modèle géo-paramétrique fondé sur les courbes de Bézier qui permet, comme pour le modèle géométrique, l'isolation de composantes connexes, et ce, dans un temps avoisinant celui du modèle paramétrique

En toute logique, nous utilisons le modèle géo-paramétrique dans le cadre de ce travail. Il est en effet à la fois le plus abouti en terme de performances, et le mieux adapté au contexte des travaux présents, notamment car il est possible de l'utiliser dans une application temps réel. Pour une analyse formelle de l'outil (en particulier des courbes de Bézier), ainsi qu'une étude fine de chacune de ses composantes, nous renvoyons le lecteur à la thèse de Pauline Merveilleux [Merveilleux 2013]. Nous nous concentrons sur ce qui importe dans notre contexte, à savoir, ses possibilités, ses performances, ainsi qu'une vue globale de la façon dont il procède.

### 2.4.3 Vue d'ensemble de l'algorithme

L'algorithme 1 propose une vue d'ensemble du processus qui conduit de l'image brute au résultat final.

---

**Algorithme 1** Extraction géo-paramétrique de l'espace libre

---

**Entrées :** Images omnidirectionnelles  $I_t$  et  $I_{t-1}$  acquises aux instants  $t$  et  $t - 1$

**Sorties :** Frontières de l'espace libre et composantes connexes

***Phase d'initialisation :***

- 1- Détection des contours sur  $I_t$  (Canny)
- 2- Initialisation du contour actif sur  $I_t$

***Phase d'expansion :***

**jusqu'à convergence, répéter :**

- 1- Déplacement du contour actif sous l'effet d'une énergie ballon
- 2- Recherche d'auto-intersection et isolation des composantes connexes
- 3- Mesure de distance entre les points et ajustement si nécessaire

***Détection et suppression des faux obstacles :***

- 1- Recherche et mise en correspondance des points d'intérêt au sein des composantes connexes dans les images  $I_t$  et  $I_{t-1}$
  - 2- Triangulation et estimation de l'altitude des points
  - 3- Suppression des composantes connexes dont les points appariés sont tous d'altitude 0
-

### 2.4.3.1 La phase d'initialisation

La phase d'initialisation s'articule autour de deux points précis : l'initialisation du contour actif et l'extraction des contours dans l'image.

Le premier est dans le cas général une tâche critique, qui est ici rendue triviale par le fait que le centre du robot est aligné avec la verticale du capteur. Le robot se projette donc au centre de l'image. Le contour actif est ainsi initialisé sous la forme d'un cercle qui englobe la zone aveugle du capteur. Cette dernière correspond au reflet de la caméra sur le miroir, et masque la projection du robot. En choisissant le rayon de ce cercle suffisamment petit, et sous réserve que la distance du robot aux obstacles soit toujours suffisante, les points appartiennent à l'espace libre proximal, et porteront donc l'information photométrique caractérisant ce dernier.

Les contours sont quant à eux extraits à partir d'un détecteur de Canny-Deriche [Canny 1986, Deriche 1987] pour lequel les deux seuils sont fixés automatiquement à partir de la médiane photométrique de l'image. De cette façon, ils sont déterminés à chaque instant en fonction du contenu image, ce qui rend la méthode tolérante (dans une certaine mesure) à des changements dans les conditions d'éclairage.

### 2.4.3.2 La phase d'expansion

Une fois l'initialisation effectuée, le contour actif entre dans une phase d'expansion qui ne s'achève que lorsqu'une proportion suffisante du contour (en l'occurrence 95%) a atteint une frontière au sens du gradient. A chaque itération, l'expansion se joue en trois actes :

- d'abord, le contour actif "gonfle" sous l'effet d'une énergie ballon [Cohen 1991] là où il n'est pas déjà en contact avec un gradient photométrique.
- la densité des points qui le discrétisent est ensuite contrôlée, et, si nécessaire, ajustée, afin d'avoir en permanence une précision suffisante.
- enfin, le contour actif est analysé dans le but de détecter une éventuelle auto-intersection. Si elle existe, une composante connexe (un obstacle inclus dans l'espace libre) est isolée par un processus de *flip*.

Ces trois étapes forment le coeur de la méthode, et le lecteur est vivement encouragé à se référer à [Merveilleux 2013] pour plus de détails.

### 2.4.3.3 L'élimination des faux obstacles

Au terme de la phase d'expansion, l'algorithme est en possession d'un contour actif principal, qui épouse les frontières de l'espace navigable, et d'une collection de composantes connexes qui entourent les objets inclus dans l'espace libre. Certains d'entre eux sont de faux obstacles : il s'agit de zones présentant des différences photométriques par rapport au sol, sans être réellement des obstacles (marquages au sol, tâches, ...). Une phase de filtrage est donc appliquée à chacune des composantes connexes. L'idée est de trianguler les points d'intérêt détectés dans chacune d'entre elles sur deux images consécutives afin d'en évaluer l'altitude. Dans le cas où au

moins l'un d'entre eux a une altitude non nulle, la composante connexe est considérée comme un obstacle à part entière. Dans le cas contraire, elle est incorporée à l'espace libre.

#### 2.4.4 Résultats

##### 2.4.4.1 Quelques résultats

Une validation expérimentale riche étant disponible dans le travail original [Merveilleux 2013], le but n'est pas ici de prouver le bien-fondé de la méthode. Toutefois, afin d'illustrer les capacités de l'algorithme, et de préparer le lecteur à la suite du mémoire, nous présentons quelques espaces libres extraits pour des topologies et des conditions expérimentales variées (Figures 4.5).

##### 2.4.4.2 Les limites du critère photométrique

En parallèle, et dans le souci de justifier certains choix réalisés dans les chapitres suivants, nous présentons également quelques situations qui mettent en évidence les limites de la méthode. En particulier, lorsque le gradient sol / obstacle n'est pas assez prononcé, il arrive que ce dernier soit incorporé à l'espace libre. À l'inverse, une composante connexe trop proche des frontières de l'espace navigable ne peut être totalement isolée, et l'ampute donc d'une partie de sa substance. Ces phénomènes sont présentés Figure.2.9.

#### 2.4.5 Discussion

Les résultats proposés juste avant mettent en évidence un certain nombre de limites. Outre les erreurs de segmentation majeures, le contour actif est, même dans un cas idéal, sujet à un bruit de haute fréquence lié aux approximations du détecteur de contour. Pour y remédier, plusieurs stratégies seraient envisageables. On pourrait par exemple remplacer le détecteur de Canny-Deriche par une solution plus performante. On pourrait également enrichir l'information photométrique en ajoutant d'autres modalités de perception (une seconde caméra pour faire de la stéréovision, un télémètre laser,...). Aucune des stratégies existantes ne permettrait toutefois de solutionner l'ensemble des problèmes, donc de produire un espace libre dans l'image parfaitement en adéquation avec la réalité physique de la scène perçue. Nous prenons le parti dans ce travail de nous satisfaire des résultats de la méthode géo-paramétrique, tout en gardant à l'esprit leurs limites afin de les prendre en considération lors de la construction des différentes représentations.

## 2.5 Conclusion

Nous possédons à l'issue de chapitre l'ensemble des outils nécessaires à la construction des représentations de l'environnement qui constituent le coeur du travail. Tout d'abord, nous nous sommes intéressés à l'unique modalité de perception



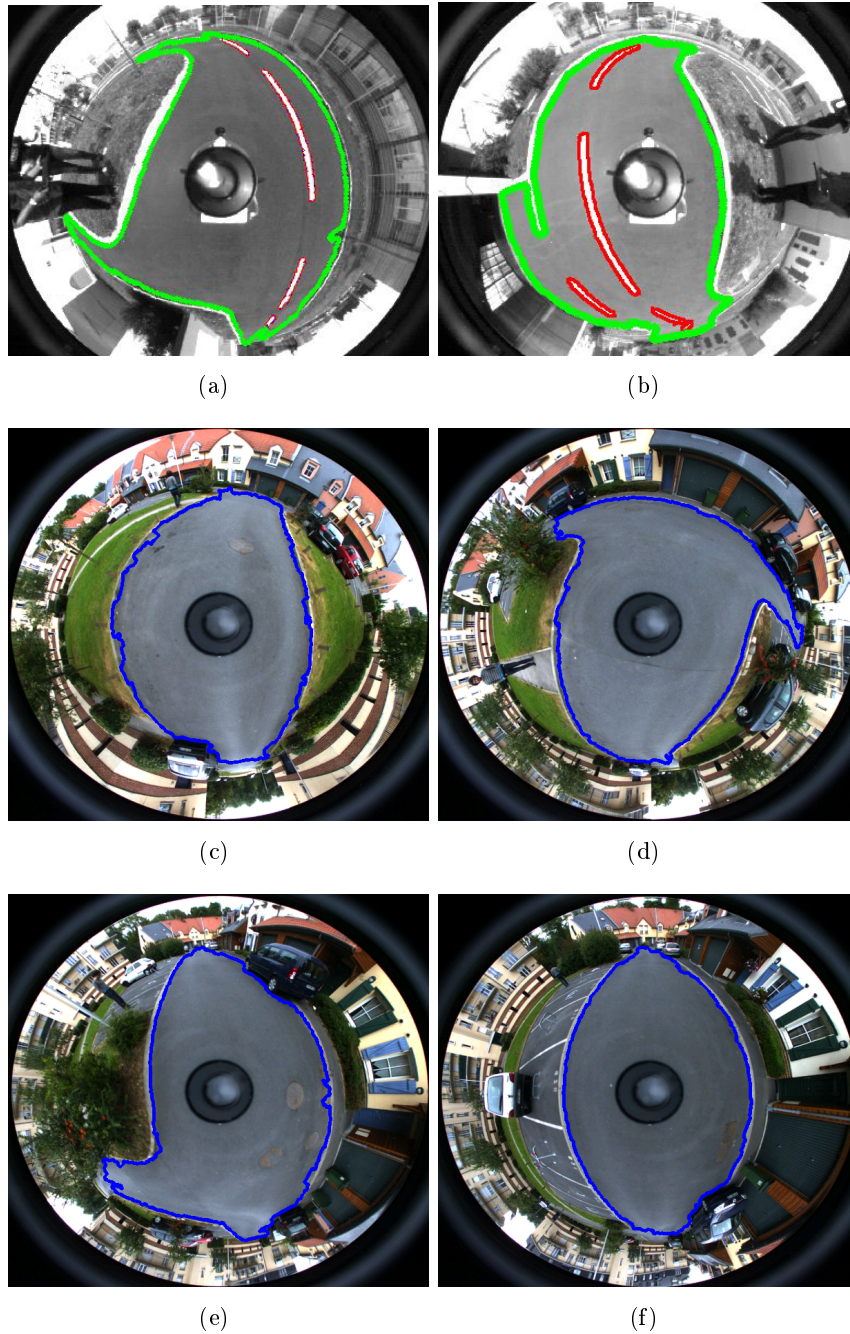


FIGURE 2.8 – Exemples de résultats obtenus avec la méthode géo-paramétrique. Ces images ont été tirées pour (a) et (b) du travail original [Merveilleux 2013] (mettant en évidence la séparation des composantes connexes), et pour (c) à (f) d'un scénario réalisé au cours de cette thèse.

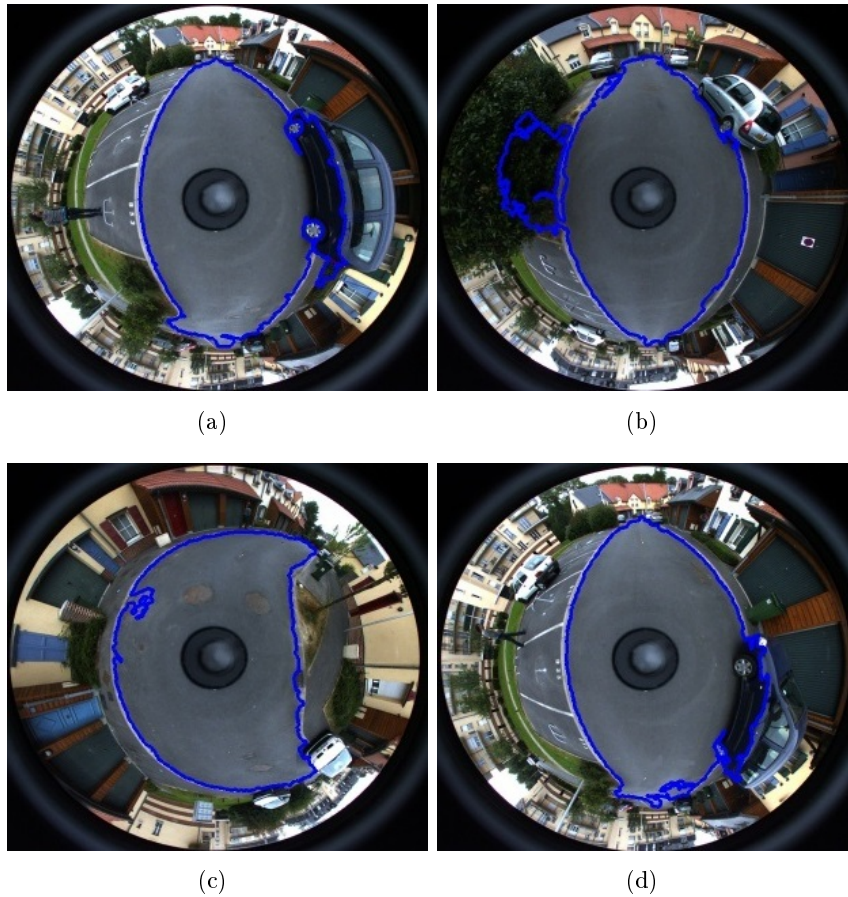


FIGURE 2.9 – Exemples de situations où l’algorithme est mis en défaut. (a) (b) la transition entre espace libre et obstacle est par endroit trop lente, permettant à l’espace libre de s’infiltrer là où il ne devrait pas. (c) la composante connexe (bouche d’égout) est trop proche des frontières de l’espace libre : elle y est intégrée. (d) les reflets sur la route rendent l’information photométrique erronée.

à disposition du robot : la vision catadioptrique. En présentant les équations qui traduisent la projection de la scène vers le plan image, nous avons introduit la notion de sphère d’équivalence qui sera largement utilisée dans la suite du mémoire. L’algorithme de segmentation de l’espace libre présenté dans la seconde section est quant à lui la pierre angulaire des chapitres à venir, puisqu’il permet de donner, en temps réel, une information de structure sur la scène observée.



# Topologie locale par calcul d'axe médian

---

## Sommaire

<b>3.1</b>	<b>Introduction</b>	<b>21</b>
<b>3.2</b>	<b>L'axe médian</b>	<b>22</b>
3.2.1	Première approche et formalisations existantes	22
3.2.2	Propriétés fondamentales	24
3.2.3	Axe médian, squelette et transformée en axe médian	25
<b>3.3</b>	<b>Etat de l'art</b>	<b>26</b>
3.3.1	Extraction d'axe médian	26
3.3.2	Filtrage des branches parasites	28
3.3.3	Cadres applicatifs	29
3.3.4	Discussion	30
<b>3.4</b>	<b>Le Delta Medial axis</b>	<b>30</b>
3.4.1	Définition des objectifs	30
3.4.2	Formulation	31
3.4.3	Implémentation et optimisations	32
3.4.4	Résultats expérimentaux	38
3.4.5	Première application : cartographie topologique d'une vue aérienne	43
<b>3.5</b>	<b>Adaptation aux images omnidirectionnelles</b>	<b>46</b>
3.5.1	Distorsions et axe médian	46
3.5.2	Projection inverse	48
3.5.3	Vers une métrique adaptée	49
3.5.4	Formalisation du ODMA	49
3.5.5	Réglage automatique de $\tilde{\delta}$	50
3.5.6	Quid de l'implémentation?	51
3.5.7	Résultats	53
<b>3.6</b>	<b>Conclusion</b>	<b>55</b>

---

## 3.1 Introduction

Nous proposons maintenant d'entrer dans le vif du sujet en présentant la première représentation locale. La segmentation de l'espace libre sépare l'image en

deux régions distinctes. Bornée par la position finale du contour actif, se trouve la projection de l'espace navigable, qui n'a que peu d'intérêt en terme d'information photométrique (car globalement homogène). Considérée en tant que forme, c'est à dire définie par un masque binaire tel que présenté Fig.3.1(b), elle porte en revanche les propriétés topologiques de la scène observée. Ce chapitre développe une représentation spatiale capable de capturer ces propriétés.



(a)



(b)

FIGURE 3.1 – a) Image omnidirectionnelle acquise par le robot et qui constitue sa seule information perceptive de la scène qui l'entoure. b) Masque définissant le sol dans l'image, obtenu après application du module de segmentation de l'espace libre. A noter que toute notion photométrique a disparu, et que seules les dimensions et la forme du masque sont conservées.

Après avoir introduit, défini et étudié la notion d'axe médian, nous nous intéresserons au *Delta Medial Axis* [Marie 2013c]. Cet algorithme extrait le squelette d'une forme définie dans un espace discret de dimension 2. Afin de garantir suffisamment de stabilité face aux éventuels bruits de segmentation et de discrétisation, il intègre un critère de filtrage original qui offre des performances supérieures à ceux utilisés dans les méthodes de l'état de l'art. Dans un deuxième temps, et après avoir souligné sa nécessité, une version adaptée à la géométrie du capteur est proposée. Elle permet de calculer non plus l'axe médian de la projection de l'espace libre, mais plutôt la projection de l'axe médian de l'espace libre, ce qui, comme nous le verrons, est fondamentalement différent.

## 3.2 L'axe médian

### 3.2.1 Première approche et formalisations existantes

Le premier à introduire la notion d'axe médian est Harry Blum [Blum 1964], qui, en 1964, est à la recherche de ce qu'il considère être un "bon" descripteur de forme. En particulier, dans un contexte biologique (étude des chromosomes), il a besoin

d'une représentation compacte capable de capturer les propriétés topologiques de la forme étudiée. Il présente alors le *medial axis* (axe médian) en utilisant une analogie qu'il décrit lui-même comme intuitive : l'analogie du feu de prairie.

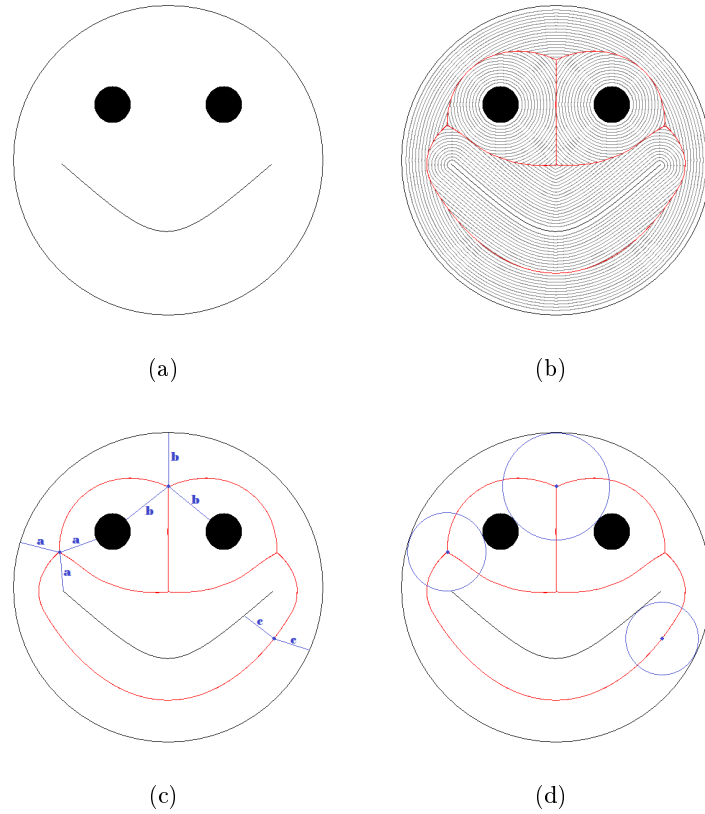


FIGURE 3.2 – Illustration des différentes formulations de l'axe médian (en rouge). (a) Forme étudiée. (b) Axe médian en tant que lieu des collisions entre fronts d'onde. (c) Axe médian, composé des points ayant une projection de cardinal supérieur ou égal à 2. (d) Axe médian vu comme l'union des centres des disques inscrits maximaux

Soient  $\mathcal{X}$  une "prairie", c'est-à-dire une forme contenue dans le plan continu  $\mathbb{R}^2$ , et  $\partial\mathcal{X}$  ses frontières. En supposant qu'à un instant  $t = 0$ , un incendie d'intensité uniforme se déclenche en tout point de  $\partial\mathcal{X}$ , le feu va, de proche en proche et à vitesse constante, envahir peu à peu l'ensemble de la forme. L'axe médian peut alors être décrit comme l'ensemble des points de  $\mathcal{X}$  qui sont le lieu d'une collision entre deux fronts différents de l'incendie (voir Fig.3.2.b).

De par la nature uniforme de la propagation des fronts de "l'incendie", une seconde formulation est également suggérée par Blum. En effet, la collision intervient nécessairement à équidistance de l'origine de chaque front, qui correspondent aux points de  $\partial\mathcal{X}$  les plus proches (faute de quoi une autre collision aurait déjà eu lieu en ce point). En dénommant projection de  $\mathbf{x}$  l'ensemble des points de  $\partial\mathcal{X}$  qui se trouvent à distance minimale de  $\mathbf{x}$ , l'axe médian de  $\mathcal{X}$  peut donc être vu comme l'ensemble

des points  $\mathbf{x} \in \mathcal{X}$  dont la projection a un cardinal d'au moins 2 (voir Fig.3.2.c). Le lien avec la première formalisation est immédiat puisqu'un cardinal de 1 implique l'absence de collision entre deux fronts d'ondes, et désigne donc un point qui n'appartient pas à l'axe médian tel que formalisé précédemment.

Une autre formulation usuelle est suggérée par Pfaltz et Rosenfeld [Pfaltz 1967]. Elle utilise cette fois la notion de disque inscrit maximal : Soit  $\mathcal{D}$  un disque tel que  $\mathcal{D} \subset \mathcal{X}$ .  $\mathcal{D}$  est dit maximal s'il n'existe aucun autre disque  $\mathcal{D}' \subset \mathcal{X}$  tel que  $\mathcal{D} \subset \mathcal{D}'$ . L'axe médian de  $\mathcal{X}$  peut alors être vu comme l'ensemble des centres des disques maximaux. Clairement, pour qu'un disque soit maximal, il doit être en contact avec  $\partial\mathcal{X}$  en au moins deux points (voir Fig.3.2.d), c'est-à-dire que sa projection doit avoir un cardinal d'au moins 2. L'équivalence avec la formalisation précédente est donc établie [Wolter 1992].

### 3.2.2 Propriétés fondamentales

Dans son article, Blum ne réalise qu'une étude générale des propriétés fondamentales de l'axe médian, et se contente d'observations fondées sur des exemples. Depuis, de nombreux auteurs se sont intéressés à ses propriétés dans le cas continu [Riley 1964, Mat 1988, Mattioli 1993, Schmitt 1993]. Les plus notoires sont les suivantes :

- **Positionnement** : l'axe médian se situe le long des axes de symétrie de la forme étudiée et peut donc être vu comme son squelette.
- **Finesse** : il est généralement admis que l'axe médian est fin, c'est-à-dire d'épaisseur nulle. Pour le cas d'une forme définie dans un espace à deux dimensions, cette propriété se traduit par le fait que l'axe médian a une structure de graphe, ce qui sera d'importance dans la suite du mémoire.
- **Invariance aux rotations et translations** : en notant  $g$  une rotation ou translation appliquée à  $\mathcal{X}$ , on a  $AM(g(\mathcal{X})) = g(AM(\mathcal{X}))$
- **Homotopie** : l'axe médian est homotopique à la forme qu'il caractérise. Il existe donc une transformation continue qui permet de passer de l'un à l'autre. Une forme connexe a donc obligatoirement un axe médian connecté.
- **Stabilité** : l'axe médian est un sous-ensemble de  $\mathbb{R}^2$ . Il possède donc son propre axe médian. La notion de stabilité sous-entend ici que  $AM(AM(\mathcal{X})) = AM(\mathcal{X})$
- **Réversibilité** : l'axe médian d'une forme ne permet pas de la reconstruire. Il est néanmoins facile de prouver que chaque point de  $\mathcal{X}$  appartient à au moins un disque inscrit maximal. En notant  $r(\mathbf{x})$  le rayon du disque inscrit maximal pour tout  $\mathbf{x}$  de  $AM(\mathcal{X})$ , on construit la transformée en axe médian  $TAM(\mathcal{X}) = \{(\mathbf{x}, r(\mathbf{x})), \mathbf{x} \in AM(\mathcal{X})\}$ , qui quant à elle est inversible. La forme  $\mathcal{X}$  est obtenue par l'union des disques inscrits maximaux décrits par  $TAM(\mathcal{X})$ .
- **Semi-continuité** : la dernière des propriétés de l'axe médian est également son plus gros défaut. La transformation qui lie une forme à son axe



médian n'est que semi-continue. Une variation mineure de la forme (au sens de la distance de Hausdorff) peut induire un changement majeur dans l'axe médian obtenu. Considérons par exemple l'axe médian d'un disque parfait (Figure.3.3(a)). Il correspond à un point placé en son centre, ce dernier se trouvant à équidistance de tous les points du cercle. Mais si une déformation mineure apparaît sur les frontières du disque (figure.3.3(b)), le squelette obtenu devient alors un segment reliant le centre à cette déformation. Topologiquement parlant, la différence est importante. Malgré une ressemblance forte entre les formes, une mise en correspondance fondée sur l'utilisation de leurs axes médians devient un vrai challenge. Le constat s'aggrave encore lorsque de nouvelles déformations s'ajoutent à la première, puisqu'elles provoquent l'apparition de nouvelles branches (figure.3.3(c)).

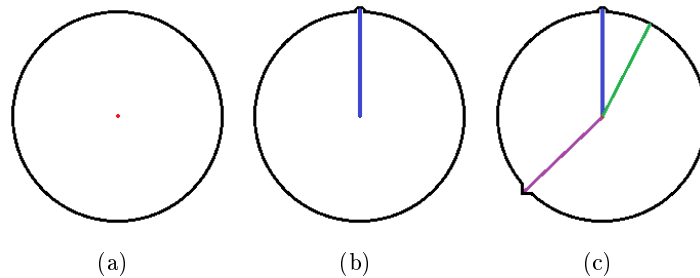


FIGURE 3.3 – Axes médians obtenus dans le cas où la forme est (a) un disque parfait, (b) un disque avec un unique défaut et (c) un disque avec plusieurs défauts. Des petits défauts dans la forme impliquent clairement des changements majeurs dans l'axe médian obtenu.

### 3.2.3 Axe médian, squelette et transformée en axe médian

Dans le travail original de Blum, le terme squelette n'apparaît pas. Ce n'est que plus tard que Pfaltz et Rosenfeld [Pfaltz 1967] l'introduisent. En l'occurrence, ils le substituent purement et simplement : l'axe médian de Blum n'est autre que le squelette de Pfaltz. Depuis, il n'existe aucun consensus concernant l'utilisation de ces deux termes. Ainsi :

- Dans [Dominique 1995], l'axe médian est défini comme la version discrète du squelette, ce dernier étant formulé à partir des disques inscrits maximaux.
- Dans [Chazal 2005], les notions respectent les définitions originelles : l'axe médian est défini en terme de cardinal des projections tandis que le squelette est l'union des centres des disques inscrits maximaux (il prouve d'ailleurs que les deux notions ne sont pas strictement équivalentes).
- Dans [Chaussard 2011], le terme squelette est utilisé dans un contexte discret pour désigner un axe médian homotopique à la forme étudiée.



- Enfin, de nombreux auteurs [Hesselink 2007, Couprie 2007] utilisent indifféremment les deux termes.

Nous faisons ce dernier choix dans ce mémoire. Par extension, le processus de calcul d'axe médian est appelé squelettisation. Pour rappel, nous utilisons également l'appellation "Transformée en axe médian", notée TAM, pour désigner la fonction qui associe à chaque point de l'axe médian sa distance au bord.

### 3.3 Etat de l'art

La littérature ayant trait à l'axe médian est très fournie, et peut être classifiée en trois catégories principales :

- Son calcul : les définitions proposées par Blum et Pfaltz ont donné naissance à toute une série de solutions qui permettent l'extraction effective de l'axe médian.
- Son filtrage : le problème de la semi-continuité soulevé précédemment est un soucis majeur dans beaucoup de cadres applicatifs. Certains travaux proposent donc des solutions permettant d'en limiter les effets.
- Son utilisation : outre le but pour lequel Blum l'a élaboré, l'axe médian trouve son utilité dans bon nombre de cadres applicatifs aux enjeux différents.

#### 3.3.1 Extraction d'axe médian

Une partie de la littérature s'intéresse à l'axe médian tel qu'initialement défini, c'est-à-dire dans des espaces continus (voir [Siddiqi 2008] pour un état de l'art sur le sujet). Néanmoins, dans la majorité des applications, et c'est le cas ici, on possède plutôt une version approximée de la forme étudiée, qui est définie dans un espace discret. Dans le cas 2D, il s'agit le plus souvent d'un masque binaire porté par une image. Le challenge est alors de produire à partir de cette forme discrète un axe médian qui soit une approximation fidèle à son homologue continu. Les définitions suggérées par Blum et Pfaltz ne sont pourtant pas facilement transposables, notamment du fait qu'il est loin d'être simple de préserver les propriétés fondamentales décrites juste avant.

De nombreux travaux traitent de cette problématique, et proposent des solutions qui peuvent être réparties suivant quatre grandes familles :

- Les approches basées morphologie mathématique
- L'utilisation du diagramme de Voronoï
- La simulation du feu de prairie
- La recherche de singularités sur la carte des distances

Dans les **approches basées morphologie mathématique** [Lam 1992, Liu 2011, Zhou 1995, She 2009], les points de la forme sont supprimés tour à tour par un processus itératif qui, la plupart du temps, utilise un critère de voisinage (comme le plus petit ensemble non érodable [Ronse 1988], les points simples [Bertrand 1995]

ou encore les noyaux critiques [Bertrand 2008]). Ces solutions garantissent généralement la connectivité du résultat final, et permettent donc dans leur quasi-totalité de conserver la propriété d'homotopie. En revanche, ni la finesse ni le bon positionnement de l'axe médian ne sont garantis.

Le second type de solution est fondé sur l'**utilisation des diagrammes de Voronoï** [Bai 2007, Ogniewicz 1995, Mayya 1996]. L'idée est d'échantillonner les frontières de la forme par un ensemble de points pour lesquels les régions de Voronoï sont évaluées. De là, les arêtes issues de deux échantillons directement voisins sont supprimées, et le résultat obtenu est une approximation de l'axe médian. Plus le nombre de points échantillonnant la forme est élevé, plus le résultat est précis. Néanmoins, l'échantillonnage des frontières de la forme est une étape critique et déterminante pour le résultat final, ce qui rend ces algorithmes généralement coûteux en temps de calcul. De plus, le résultat n'est qu'une approximation de l'axe médian, et ne vérifie donc généralement pas la propriété de positionnement.

D'autres méthodes partent de la définition originelle de Blum, à savoir l'**analogie du feu de prairie**. Elles utilisent alors différents outils mathématiques ou analytiques pour la simuler. Dans [Siddiqi 2002], le formalisme d'Hamilton-Jacobi sert à simuler la propagation du front d'onde et permet de détecter analytiquement les singularités (les lieux de collision) par la résolution d'équations différentielles partielles. Dans [Kimmel 1995, Rumpf 2002], ce sont les courbes de niveau (level set) qui servent de base pour simultanément calculer la carte des distances et y localiser les singularités. Dans [Leymarie 1992], un contour actif initialisé sur les frontières de la forme subit un processus de rétractation jusqu'à atteindre l'axe médian convoité. Enfin, dans [Ahuja 1997, Chuang 2000], l'axe médian est défini par les minima d'un champ potentiel limité par les frontières de l'objet (où la valeur y est maximale). Ces méthodes offrent en général des résultats de qualité qui préservent la connectivité du résultat final (donc la propriété d'homotopie) ainsi que son positionnement et sa finesse. Elles sont en revanche très coûteuses en temps de calcul et difficiles à mettre en place.

La dernière catégorie de méthodes commence par déterminer en chaque point de la forme **la distance au bord et/ou la projection** (le/les point(s) frontière(s) à distance minimale). Ce résultat est ensuite utilisé pour localiser les points appartenant à l'axe médian. Dans [Borgefors 1989], les maxima locaux dans la carte des distances sont détectés et utilisés pour d'une part extraire l'axe médian d'une forme, et d'autre part permettre sa reconstruction. Dans [Arcelli 1993, Ge 1996], les auteurs proposent une solution pour localiser les disques inscrits maximaux à partir de la carte des distances, et donc, par extension, les points appartenant à l'axe médian (le résultat obtenu étant discontinu, une phase de connexion est ajoutée). Dans [Hesselink 2007, Couprie 2007, Chaussard 2011], c'est cette fois les projections qui sont utilisées afin de déterminer si un point répond à un critère d'appartenance à l'axe médian (typiquement, si les deux projections sont suffisamment éloignées l'une de l'autre). Ces méthodes sont de manière générale les plus rapides à calculer, la carte des distances pouvant être extraite en temps réel ([Hirata 1996]). Le résultat obtenu est néanmoins très souvent instable, et nécessite une étape supplémentaire

de filtrage, parfois directement incluse dans le processus d'extraction d'axe médian ([Hesselink 2007, Chaussard 2011]).

### 3.3.2 Filtrage des branches parasites

Comme expliqué précédemment, la transformation qui associe une forme à son axe médian n'est que semi-continue : une petite déformation de la forme peut induire une variation importante de l'axe médian. Dans de nombreux contextes (incluant le nôtre, voir Figure 3.4), il s'agit d'un problème majeur, qui a donné lieu à tout un ensemble de solutions permettant de contrôler l'apparition des branches parasites.

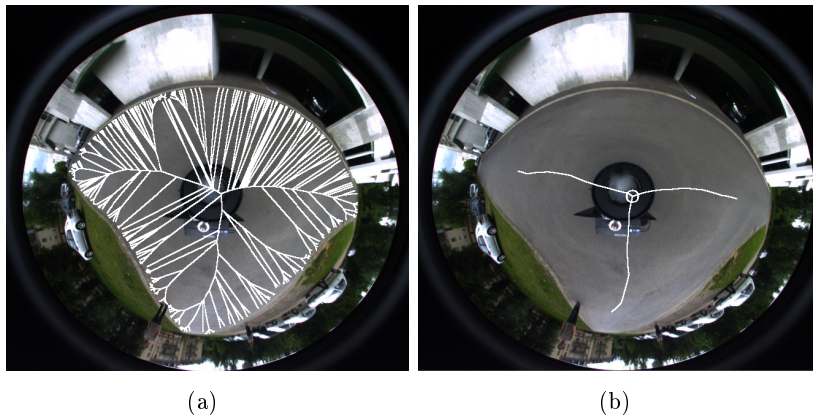


FIGURE 3.4 – (a) Axe médian obtenu à partir d'un algorithme n'intégrant pas de processus de filtrage. (b) Avec filtrage, résultat envisageable et surtout utilisable par le robot.

Une première famille de filtrage [Pizer 1987, Mokhtarian 1992, Rumpf 2002] cherche à travailler sur la forme étudiée afin d'en retirer les défauts mineurs. Cela se traduit par une opération de lissage du contour qui permet de supprimer certaines branches liées à des déformations négligeables. Cette approche présente deux défauts majeurs : d'abord, l'axe médian de la forme filtrée n'est, la plupart du temps, pas centrée sur la forme étudiée, et ne vérifie donc pas la propriété de positionnement présentée plus tôt. Ensuite, le processus de lissage peut entraîner la suppression d'éléments topologiques importants, et par conséquent induire la perte de la propriété d'homotopie.

La seconde approche évalue l'axe médian de la forme brute, puis attribue à chaque point  $y$  appartenant un degré d'importance fondé sur un critère qui dépend de la méthode. Un seuil fixé permet ensuite de ne conserver que les points jugés suffisamment pertinents. La plus simple des approches [Davies 1981] considère le rayon du disque inscrit maximal en chaque point. Le résultat d'un tel processus de filtrage est le raccourcissement de l'ensemble des branches de l'axe médian (notamment les plus importantes), d'une longueur égale à la valeur du seuil. D'autres critères plus élaborés ont depuis été proposés. Commençons par

la *bisector function* de Meyer [Meyer 1979], très largement utilisée et étudiée [Talbot 1992, Malandain 1998, Attali 2001, Couprie 2007]. Elle associe à chaque point de l'axe médian, l'angle  $\theta$  formé par ce dernier (le sommet) et les deux points des frontières de la forme qui se trouvent à distance minimale. Ce critère offre l'avantage d'être robuste aux variations d'échelle au sein de la forme, mais ne garantit pas la connectivité du squelette obtenu. Dans [Chazal 2005], une étude théorique est réalisée sur un autre paramètre de filtrage, noté  $\lambda$ , qui correspond au rayon du plus petit cercle englobant l'ensemble des points frontières à distance minimale. Le  $\lambda$ -medial axis ainsi produit est homotopique à la forme étudiée à la condition que la valeur  $\lambda$  soit choisie inférieure à tout élément topologique important de la forme. Ces travaux sont repris dans [Chaussard 2011] pour le cas discret. En outre, ils sont à rapprocher des travaux de Hesselink [Hesselink 2007] qui présente quant à lui le  $\gamma$ -Integer Medial Axis. La valeur  $\gamma$ , utilisée correspond cette fois à la distance entre les deux points frontières les plus proches. Dans [Coeurjolly 2007], un critère plus complexe est utilisé : l'idée est de déterminer la portion de la forme exclusive au disque inscrit maximal de chaque point. C'est-à-dire la surface du disque inscrit maximal qui n'appartient au disque inscrit maximal d'aucun autre point formant l'axe médian.

Dans une catégorie un peu à part, on peut enfin citer le *scale medial axis* [Miklos 2010, Giesen 2009, Postolski 2013] qui procède en trois temps. D'abord, la transformée en axe médian de la forme brute est calculée. En chacun de ses points, la distance au bord est multipliée par un coefficient amplificateur. A partir de cette transformée en axe médian modifiée, la forme "gonflée" est reconstruite. Enfin, l'axe médian de cette nouvelle forme est estimé, et sert d'approximation à celui de la forme bruitée.

### 3.3.3 Cadres applicatifs

Dès 1978, Blum [Blum 1978] propose une application concrète de l'axe médian en s'en servant comme outil pour la classification de formes. Depuis, son utilité a été démontrée dans de nombreux cadres différents.

Il est tout d'abord largement utilisé dans un contexte de caractérisation et reconnaissance de formes [Sebastian 2005, Siddiqi 1998, Eede 2006]. Dans [Moayer 1975], une approche l'utilisant est suggérée pour la reconnaissance d'empreintes digitales. Dans [Kegl 2002], c'est de la reconnaissance de chiffres manuscrits qui est proposée.

D'autres travaux tirent partie de la propriété de réversibilité de la transformée en axe médian dans un contexte de compression de données. Dans [Brandt 1991], et [Pujar 2010], deux algorithmes de compression pour images binaires sont proposés. Dans [Dupont 2003], c'est cette fois une solution pour transférer des formes tridimensionnelles qui est proposée. L'idée est de segmenter la forme en un ensemble de cônes et boules obtenues par la transformée en axe médian, permettant ainsi la transmission de la forme par morceau.

Enfin, dans un contexte de robotique mobile, d'autres travaux utilisent l'axe médian pour extraire en amont ou en ligne la topologie de l'environnement ex-

ploré (généralement sous la forme de son diagramme de Voronoi généralisé). Dans [Masehian 2003], l'axe médian est calculé à partir des modalités de perception disponibles et sert de base à un processus de navigation autonome et réactif. Dans [Farhan 2011], un graphe topologique est construit en appliquant un algorithme d'extraction d'axe médian sur une représentation 2D (*blueprint*) de l'environnement. La carte cognitive obtenue sert ensuite de support pour déterminer le chemin le plus court entre deux positions de l'environnement. Dans [Garrido 2011], un algorithme de squelettisation est appliqué sur une grille d'occupation et permet l'utilisation d'une méthode de Fast Marching pour la planification de trajectoire.

### 3.3.4 Discussion

Les solutions qui viennent d'être présentées, que ce soit pour l'extraction de l'axe médian ou pour son filtrage, font pour l'immense majorité des concessions par rapport aux propriétés fondamentales de l'outil. Quand ce n'est pas le cas, elles impliquent des temps de calcul très importants. Lorsqu'une application fondée sur l'utilisation d'axe médian est élaborée, elle vise en général des objectifs bien précis. Il est donc indispensable de cibler ces besoins.

Ainsi, une application se servant de l'axe médian en tant que descripteur de forme nécessite que sa topologie reste stable, même en présence de bruit de segmentation. Dans ce contexte, la connectivité est un critère primordial, de même que l'élimination des branches non désirées. En contrepartie, l'algorithme pourra généralement impliquer un temps de calcul un peu plus long, et l'axe médian ne vérifiera pas nécessairement la condition de positionnement.

A l'inverse, une méthode visant des objectifs de compression de données ne verra pas la question de la connectivité comme une priorité, puisqu'à partir du moment où l'ensemble des cercles formés à partir de l'axe médian modélise correctement la forme, le résultat est satisfaisant. La compression sera d'ailleurs meilleure dans le cas où le nombre de points qui composent l'axe médian est minimal. En revanche, le positionnement est dans ce cas un critère primordial, puisque la reconstruction est basée sur des cercles dont le centre doit être précisément situé, faute de quoi la forme reconstruite différera de l'originale.

## 3.4 Le Delta Medial axis

### 3.4.1 Définition des objectifs

La représentation que nous cherchons à construire a pour vocation de permettre au robot de se guider dans l'environnement, mais également de lui fournir un support sur lequel construire une carte topologique. L'outil développé passe donc par la détermination en ligne, et en temps réel (pour garantir une navigation aussi sûre que réactive), des différentes branches du squelette de la scène observée (depuis, on le rappelle, un capteur paracatadioptrique, source d'importantes distorsions dans l'image résultante). De plus, la forme sur laquelle est appliquée l'algorithme est le

fruit de la procédure d'extraction d'espace libre présentée dans le chapitre précédent. Elle est donc susceptible d'être perturbée non seulement par le bruit d'acquisition lié au capteur, mais aussi et surtout par d'éventuelles erreurs de segmentation. Le delta medial axis a donc été élaboré afin :

- de prendre en entrée une forme définie comme un sous-ensemble de  $\mathbb{Z}^2$ ,
- d'intégrer un processus d'élagage efficace et adapté,
- de produire un axe médian connecté, correctement positionné, et d'une finesse optimale,
- de présenter une complexité linéaire (ou quasi-linéaire) et des temps de calcul compétitifs,
- et enfin de permettre, sans reformulation complète, une prise en compte des propriétés du système d'acquisition afin d'obtenir un résultat cohérent avec la scène observée.

A l'inverse, la question de la réversibilité n'est pas importante. Au contraire, du fait de l'élagage, et à moins que la forme ne soit parfaitement lisse, on est certain de ne pas être capable de la reconstruire à partir de son DMA. De même, là où certains travaux cherchent à construire un axe médian minimal (notamment dans le domaine de la compression de données), ce n'est clairement pas la préoccupation ici.

### 3.4.2 Formulation

Forts de cette analyse, nous sommes maintenant en position de proposer une formulation pour le *Delta Medial Axis*. Nous sortons dans un premier temps du cadre applicatif qui est le nôtre et considérons le cas général d'une forme  $\mathcal{X}$  définie comme un sous-ensemble de  $\mathbb{Z}^2$ , avec  $\partial\mathcal{X}$  désignant ses frontières. Pour tout point  $\mathbf{x}$  de  $\mathcal{X}$ , nous définissons la projection  $\mathcal{P}(\mathbf{x})$  comme le point de  $\partial\mathcal{X}$  qui se trouve à distance minimale de  $\mathbf{x}$ . Autrement dit :

$$\forall \mathbf{x} \in \mathcal{X}, \mathcal{P}(\mathbf{x}, \partial\mathcal{X}) = \mathcal{P}(\mathbf{x}) = \{\mathbf{y} \in \partial\mathcal{X} \mid \forall \mathbf{z} \in \partial\mathcal{X}, d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z})\} \quad (3.1)$$

Dans certains cas, plusieurs points de  $\partial\mathcal{X}$  sont candidats.  $\mathcal{P}(\mathbf{x})$  est alors choisie en respect de l'ordre lexical (similairement à ce qui est proposé dans [Hesselink 2007]). La distance correspondante, nommée distance au bord et notée  $\mathcal{D}(\mathbf{x})$ , se trouve donc définie par :

$$\mathcal{D}(\mathbf{x}, \partial\mathcal{X}) = \mathcal{D}(\mathbf{x}) = d(\mathbf{x}, \mathcal{P}(\mathbf{x})) \quad (3.2)$$

Ces deux outils sont fonction d'une métrique  $d$ , qui, dans le cas général, et sauf indication contraire, désigne la distance euclidienne :

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{Z}^2, d(\mathbf{x}, \mathbf{y}) = d_E(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}_1 - \mathbf{y}_1)^2 + (\mathbf{x}_2 - \mathbf{y}_2)^2} \quad (3.3)$$

Nous introduisons enfin le voisinage direct  $V_4(\mathbf{x})$  comme l'ensemble des points à distance unitaire de  $\mathbf{x}$  :

$$V_4(\mathbf{x}) = \{\mathbf{y} \in \mathbb{Z}^2 \mid d(\mathbf{x}, \mathbf{y}) = 1\} \quad (3.4)$$

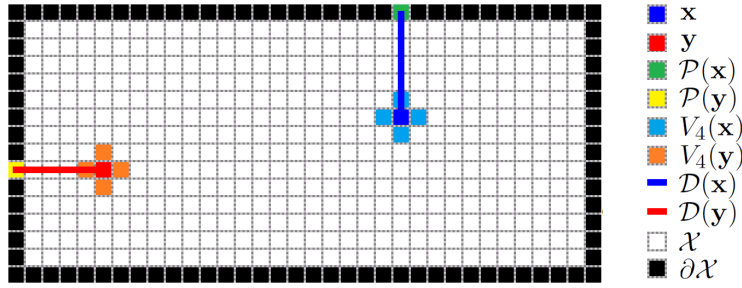


FIGURE 3.5 – Illustration des différentes notions utilisées pour définir le DMA

Ces différentes notions sont illustrées figure. 3.5.

Le *Delta Medial Axis* de  $\mathcal{X}$ , noté  $DMA(\mathcal{X})$  est alors défini comme suit :

**Définition 1** Soit  $\delta \in \mathbb{R}_+^*$ . Le  $\delta$ -medial axis de  $\mathcal{X}$  est l'ensemble des points  $\mathbf{x} \in \mathcal{X}$  qui vérifient les propriétés suivantes :

$$\exists \mathbf{y} \in \mathcal{V}_4(\mathbf{x}) \text{ t.q. } \begin{cases} \exists \mathbf{z} \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})] \mid \mathcal{D}(\mathbf{z}) \geq \delta \\ \mathcal{D}(\mathbf{x}) \geq \mathcal{D}(\mathbf{y}) \end{cases} \quad (3.5)$$

Arrêtons-nous quelques instants sur cette formulation. A l'image de ce qui est fait dans de nombreux travaux [Chaussard 2011, Couprie 2007, Hesselink 2007], l'idée est d'évaluer tour à tour chaque couple de points voisins afin de déterminer s'il existe un point de l'axe médian réel (RMA pour *Real Medial Axis*) quelque part sur le segment continu les reliant, et, si tel est le cas, de s'assurer qu'il appartient à une branche non négligeable. La première condition est la pierre angulaire de la méthode. Elle considère les projections respectives des deux points voisins, et vérifie s'il est possible d'insérer dans  $\mathcal{X}$  un disque de rayon  $\delta$  centré sur l'un des points du segment les reliant. Si tel est le cas, alors d'une part, il est certain qu'un point du RMA existe entre  $\mathbf{x}$  et  $\mathbf{y}$  (ce point, notons le  $p$  vérifiera alors  $d(p, \mathcal{P}(\mathbf{x})) = d(p, \mathcal{P}(\mathbf{y}))$ ), et d'autre part, l'algorithme estime que ce point n'appartient pas à une branche parasite au sens de  $\delta$  (qui peut être vu comme la déformation minimale sur  $\partial\mathcal{X}$  nécessaire pour générer une branche de l'axe médian). De manière induite, et contrairement à ce qui est proposé dans la littérature, cette condition prend en considération l'allure du contour séparant les deux projections, tel qu'illustré Figure 3.6.

La seconde condition est quant à elle utilisée dans la quasi totalité des travaux existants, et garantit la finesse de l'axe médian produit, puisque seul le point qui constitue la meilleure approximation du RMA est ajouté (à savoir celui qui se trouve le plus loin de  $\partial\mathcal{X}$ ). Bien sûr, dans le cas d'une équidistance, les deux points sont ajoutés.

### 3.4.3 Implémentation et optimisations

L'implémentation effective du DMA a été soignée afin d'offrir des performances temps réel. Prenant en entrée un masque binaire, l'algorithme procède en deux étapes :

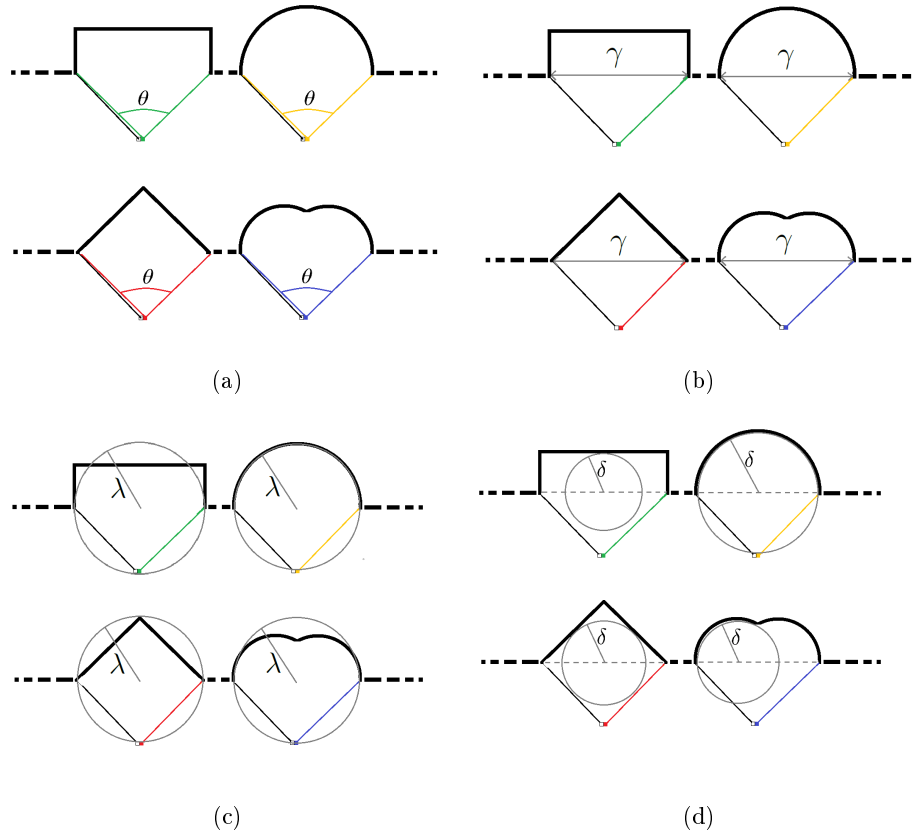


FIGURE 3.6 – Mise en évidence de l'intérêt du paramètre  $\delta$  relativement à trois méthodes de l'état de l'art, à savoir, (a) : le *Bisector Angle* [Couprie 2007], (b) : le *Gamma Integer Medial Axis* [Hesselink 2007], et (c) : le *Discrete Lambda Medial Axis* [Chaussard 2011]. Quatre topologies de contour sont considérées. Pour chacune, la valeur du paramètre associé à la méthode est illustrée. On constate que le DMA voit la valeur de son paramètre changer en fonction de la topologie, ce qui n'est pas le cas des autres méthodes.

- 1) pour chaque point  $\mathbf{x}$  de la forme  $\mathcal{X}$ , la projection  $\mathcal{P}(\mathbf{x})$  et la distance au bord  $\mathcal{D}(\mathbf{x})$  sont évaluées. L'algorithme utilisé [Hirata 1996] produit une solution exacte en deux parcours de l'image, et présente une complexité linéaire, ce qui colle parfaitement avec nos objectifs.
- 2) l'équation 3.5 est appliquée à chaque paire de pixels voisins  $\mathbf{x}, \mathbf{y}$  de  $\mathcal{X}$ . Afin de maintenir une complexité quasi-linéaire, plusieurs optimisations sont apportées, notamment concernant le segment  $[\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})]$ , qui, dans la majorité des cas, ne nécessite pas d'être étudié.

Nous nous intéressons maintenant en détail à chacune de ces deux étapes.



### 3.4.3.1 Calcul des projections et distances au bord

La première phase de l'algorithme vise à calculer pour chaque pixel de la forme sa projection et sa distance au bord. Ces deux notions sont étroitement liées, puisqu'une fois  $\mathcal{P}(\mathbf{x})$  extraite, un simple calcul de norme permet de déduire  $\mathcal{D}(\mathbf{x})$ . D'ailleurs, une partie des algorithmes cherchant à construire la transformée en distance d'une forme, c'est-à-dire la carte qui associe à chaque point sa distance au bord, utilise comme résultat intermédiaire ladite projection.

Nous choisissons dans ce travail une solution proposée dans deux travaux indépendants par Hirata [Hirata 1996] et Meijster [Meijster 2000], puis utilisée dans un contexte d'axe médian par plusieurs auteurs [Coeurjolly 2007, Hesselink 2007]. Soit  $\mathbf{x} = (i, j)$  un point défini dans le repère pixellique (avec  $i$  l'indice correspondant à la ligne, et  $j$  celui de la colonne). Sa distance au bord  $\mathcal{D}(\mathbf{x})$  (élevée au carré) peut s'écrire :

$$\mathcal{D}(\mathbf{x})^2 = \min_{(u,v) \in \partial\mathcal{X}} \{(i-u)^2 + (j-v)^2\} \quad (3.6)$$

A partir de cette formulation, un algorithme en deux étapes est suggéré. La procédure est la suivante (nous utilisons les mêmes notations que [Meijster 2000]) :

- d'abord, chaque ligne  $l$  est considérée comme un signal 1D qui permet pour chaque  $\mathbf{x} \in \mathcal{X}$ , avec  $\mathbf{x} \in l$ , d'évaluer la distance au bord 1D  $g(\mathbf{x})$  correspondante :

$$g(l, j) = \min_{(l,v) \in \partial\mathcal{X}} |j - v| \quad (3.7)$$

- Ensuite, c'est au tour des colonnes d'être parcourues, ce qui permet d'aboutir à l'extraction du carré de la distance au bord  $\mathcal{D}$  par la résolution de :

$$\mathcal{D}(i, j)^2 = \min_l ((i-l)^2 + g(l, j)^2) \quad (3.8)$$

Pour illustrer la démarche, prenons le cas d'une grille discrète simple présentée Figure 3.7(a). La première étape peut être effectuée de façon triviale par un parcours dans les deux sens de chaque ligne. L'algorithme, de complexité linéaire, est présenté Algo.2 et illustré Figure 3.7(b) et (c).

La seconde étape représente la contribution majeure de [Hirata 1996] et [Meijster 2000]. Ces deux travaux proposent un algorithme optimal permettant de la réaliser, et offrent une solution de complexité linéaire dans sa totalité.

L'idée est de remarquer que, en considérant chaque colonne  $j$  comme un signal 1D, alors,  $(i-l)^2 + g(l, j)^2$  définit une parabole dont le sommet est le point  $(l, g(l, j))$ . Les différentes valeurs de  $l$  (une par ligne) en définissent donc une collection dont l'enveloppe inférieure porte les distances au bord de chaque point de la colonne (voir Figure 3.7(f)). Reste à déterminer cette enveloppe inférieure, c'est-à-dire en chaque point, la parabole dont la valeur est minimale. Cette opération est réalisée en un seul passage pour chaque colonne, présenté dans l'Algo.3.

L'approche utilisée consiste à construire un index, qui, au fur et à mesure du parcours de la colonne, est mis à jour en intégrant les paraboles correspondant

**Algorithme 2** Distances au bord et Projections, Etape 1

**Entrées :** Masque binaire  $\mathcal{X} \cup \overline{\mathcal{X}}$  de dimensions  $m * n$

**Sorties :** Distances au bord  $g$  et projections 1D  $\mathcal{P}_{1D}$  définies pour chaque ligne  $l$

**Pour**  $l=1$  à  $m$

*Premier scan de la ligne  $l$  :*

**Si**  $(l, 1) \in \overline{\mathcal{X}}$  Alors  $g(l, 1) = 0$ ,  $\mathcal{P}_{1D}(l, 1) = (l, 1)$

**Sinon**  $g(l, 1) = \infty$ ,  $\mathcal{P}_{1D}(l, 1) = \emptyset$

**Fin si**

**Pour**  $j=2$  à  $n$

**Si**  $(l, j) \in \overline{\mathcal{X}}$  Alors  $g(l, j) = 0$ ,  $\mathcal{P}_{1D}(l, j) = (l, j)$

**Sinon**  $g(l, j) = g(l, j - 1) + 1$ ,  $\mathcal{P}_{1D}(l, j) = \mathcal{P}_{1D}(l, j - 1)$

**Fin si**

**Fin pour**

*Second scan :*

**Pour**  $j=n - 1$  à  $1$

**Si**  $g(l, j + 1) < g(l, j)$  Alors  $g(l, j) = g(l, j + 1) + 1$ ,  $\mathcal{P}_{1D}(l, j) = \mathcal{P}_{1D}(l, j + 1)$

**Fin pour**

**Fin pour**

à chaque ligne. Pour cela, la fonction  $Sep_c(s[q], i)$  est introduite. Elle permet de déterminer l'intersection entre les paraboles contenant respectivement les points  $(s[q], c)$  et  $(i, c)$  et est définie par :

$$Sep_c(s[q], i) = (i - s[q])^2 + g(i, c)^2 - g(s[q], c)^2 \mathbf{div}(2(i - s[q])) \quad (3.9)$$

Au terme de ce processus, l'index contient la liste ordonnée des paraboles utiles (d'indice  $q$ ) avec leurs sommets respectifs  $s[q]$ , ainsi que les positions  $t[q]$  où deux d'entre elles s'intersectent (voir Figure 3.7(e)). Un passage inverse dans chaque colonne permet d'utiliser cette liste afin de pouvoir enfin calculer en chaque point la distance au bord (ou plutôt son carré). Le résultat pour notre grille de référence est présenté Figure 3.7(d).

Dans les algorithmes 2 et 3, les modifications apportées par rapport à l'algorithme original sont présentées en rouge. Elles permettent d'extraire la projection  $\mathcal{P}$  en même temps que la distance au bord  $\mathcal{D}$ . Ces modifications sont mineures, et consistent simplement à localiser et à propager les points qui permettent de minimiser la distance au bord.

### 3.4.3.2 Evaluation du DMA

Pour chaque couple de points voisins  $\mathbf{x}, \mathbf{y}$ , il faut ensuite parcourir le segment formé par leur projection afin de déterminer s'il existe un point  $\mathbf{z}$  y appartenant dont

**Algorithme 3** Distances au bord et Projections, Etape 2

**Entrées :** Distance au bord 1D  $g$  et projection 1D  $\mathcal{P}_{1D}$  pour chaque point de  $\mathcal{X} \cup \overline{\mathcal{X}}$

**Sorties :** Distance au bord  $\mathcal{D}$  et projection  $\mathcal{P}$  pour chaque point du masque

**Pour**  $c=1$  à  $n$

*Troisième scan :*

$q = 0, s[q] = 0, t[q] = 0$

**Pour**  $i=2$  à  $n$

**Tant que**  $(t[q] - s[q])^2 + g(s[q], c)^2 > (t[q] - i)^2 + g(i, c)^2$

$q - -$

**Fin tant que**

**Si**  $q < 0$  Alors  $q = 0, s[q] = i$

**Sinon si**  $1 + Sep_c(s[q], i) \leq n$  Alors

$q + +, s[q] = i, t[q] = 1 + Sep_c(s[q], i)$

**Fin si**

**Fin pour**

*Quatrième scan :*

**Pour**  $i=n$  à  $1$

$\mathcal{D}(i, c) = (i - s[q])^2 + g(s[q], c)^2, \mathcal{P}(i, c) = \mathcal{P}_{1D}(s[q], c)$

**Si**  $i = t[q]$  Alors  $q - -$  **Fin si**

**Fin pour**

**Fin pour**

la distance au bord est supérieure au seuil  $\delta$  fixé. Dans le cas général, la complexité de cette étape n'est pas linéaire, puisqu'elle dépend de la longueur du segment formé par les projections. On montrera néanmoins que nous maintenons une complexité quasi linéaire en réalisant un parcours optimisé de chaque segment.

On définit un point  $\mathbf{z} \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})]$  par  $\mathbf{z} = \alpha \mathcal{P}(\mathbf{x}) + (1 - \alpha) \mathcal{P}(\mathbf{y})$ ,  $\alpha \in [0, 1]$ . Le processus d'optimisation s'articule autour de trois points majeurs, définis comme suit :

- On commence par vérifier si  $d(\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})) > 2\delta$ . Dans le cas contraire, on a  $\forall \mathbf{z} \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})], d(\mathbf{z}, \mathcal{P}(\mathbf{x})) < \delta$  ou  $d(\mathbf{z}, \mathcal{P}(\mathbf{y})) < \delta$ .  $\mathcal{P}(\mathbf{x})$  comme  $\mathcal{P}(\mathbf{y})$  appartenant à  $\partial\mathcal{X}$ , alors nécessairement  $\mathcal{D}(\mathbf{z}) < \delta$ . Ce simple test permet donc d'éliminer sans même étudier le segment une majorité des points de  $\mathcal{X}$ .
- Dans le cas où cette condition se vérifie, le parcours du segment est également optimisé. D'abord, le premier point étudié est son milieu ( $\alpha = 0.5$ ), puisqu'il est dans la majorité des cas celui qui maximise la distance au bord. Un unique point étant suffisant pour affirmer l'appartenance au  $\delta$ -MA, une distance au bord de ce point supérieure à  $\delta$  valide la condition sans parcourir le segment.
- Dans le peu de cas restant, les points du segment qui sont étudiés peuvent être judicieusement sélectionnés. Soit  $L$  la longueur du segment :  $L = d(\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y}))$  et  $\mathbf{z}_t$  (défini par  $\alpha_t$ ) le dernier point étudié (et qui ne vérifie pas la condition puisque le parcours continue). Par construction, tout point

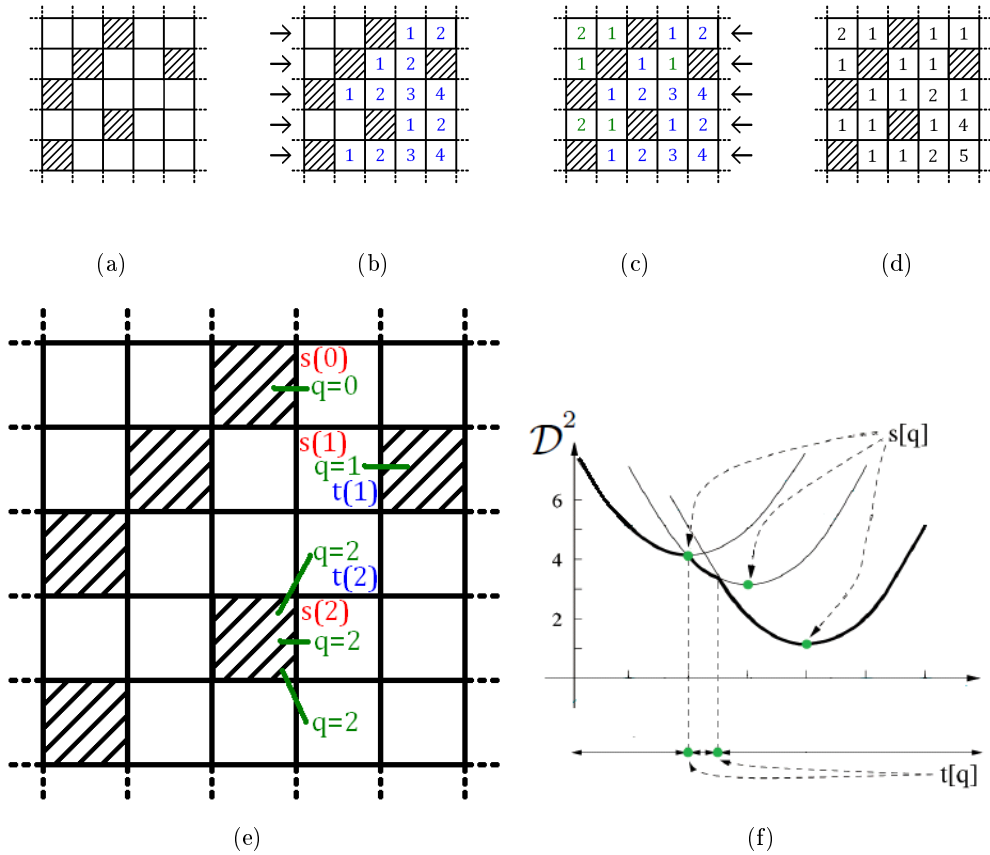


FIGURE 3.7 – (a) Grille discrète utilisée pour illustrer l’algorithme. (b) et (c) : Distances au bord 1D  $g$  après les premiers et seconds scan. (d) : Distances au bord à l’issue du quatrième scan. (e) et (f) : Illustration de  $q$ ,  $s[q]$  et  $t[q]$  sur la grille et sur les paraboles (pour la quatrième colonne).

qui se situe à distance inférieure à  $\delta - \mathcal{D}(\mathbf{z}_t)$  de  $\mathbf{z}_t$  ne peut pas non plus vérifier la condition, sa projection potentielle la plus lointaine étant  $\mathcal{P}(\mathbf{z}_t)$ , située à une distance inférieure à  $\delta$ . Alors, on pose  $\alpha_{t+1}^+ = \alpha_t + \frac{\delta - \mathcal{D}(\mathbf{z}_t)}{L}$  et  $\alpha_{t+1}^- = \alpha_t - \frac{\delta - \mathcal{D}(\mathbf{z}_t)}{L}$ , qui permettent de définir les deux points  $\mathbf{z}_{t+1}^+$  et  $\mathbf{z}_{t+1}^-$  du segment candidats pour l’itération suivante. Au départ du point central, l’idée est de partir simultanément dans les deux directions (utilisation de deux *threads*) et de s’arrêter soit lorsqu’un point vérifie la condition, soit lorsque les extrémités du segment ont été atteintes. La figure 3.8 illustre les différentes notions utilisées. A noter que le segment est dans la pratique défini dans un espace discret. Chaque valeur de  $\alpha$  choisie donne donc lieu à une interpolation pour sélectionner le point de la grille qui en constitue la meilleure approximation.

Au terme de cette seconde phase, le delta medial axis de la forme est extrait.

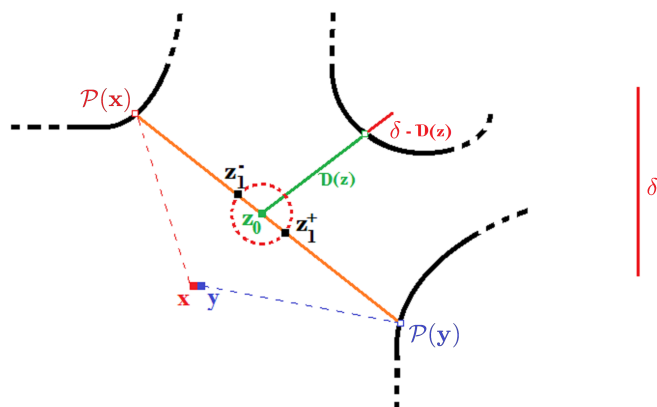


FIGURE 3.8 – Illustration du processus d'exploration de segment.

L'algorithme 4 en propose une vue d'ensemble.

### 3.4.4 Résultats expérimentaux

#### 3.4.4.1 Cadre expérimental

Dans le souci de fournir une base de comparaison pertinente, nous avons sélectionné dans la littérature deux méthodes qui se rapprochent du DMA, que ce soit en terme d'approche (utilisation des distances au bord et projections, élagage fondé sur un unique paramètre) ou de complexité algorithmique. Ces deux méthodes sont :

- Le  $\gamma$ -Integer Medial Axis [Hesselink 2007] (GIMA) qui, pour deux points voisins  $\mathbf{x}$ ,  $\mathbf{y}$ , propose un processus d'élagage fondé sur la distance entre leurs projections.
- Le Discrete $\lambda$ -Medial Axis [Chaussard 2011] (DLMA), qui, pour un point  $\mathbf{x}$  donné, s'intéresse au rayon du plus petit cercle capable d'englober les projections de chacun de ses 4 voisins.

Pour procéder à une évaluation neutre, nous utilisons la banque de 216 images de Kimia [Sebastian 2001], composée de formes binaires connues (animaux, objets courants) sujettes à un bruit de segmentation dont l'amplitude est variable.

#### 3.4.4.2 Meilleur squelette homotopique

Pour les formes de cette banque d'images, nous cherchons dans un premier temps à déterminer le meilleur squelette homotopique qu'il est possible d'obtenir. Pour cela, nous introduisons la notion de paramètre maximal. Noté respectivement  $\delta_{max}$ ,  $\gamma_{max}$  et  $\lambda_{max}$ , il désigne pour chaque point d'une forme donnée la plus grande valeur du paramètre qui permet d'inclure ce point dans l'axe médian calculé. Pour chaque  $\mathbf{x} \in \mathcal{X}$ , il est défini comme suit :

**Algorithme 4** Extraction du DMA

**Entrées :** Distance au bord  $\mathcal{D}$  et projection  $\mathcal{P}$  pour chaque point de  $\mathcal{X} \cup \overline{\mathcal{X}}$ ;  
 Paramètre d'élagage  $\delta$   
**Sorties :**  $\delta$ -medial axis  $DMA$

*Cinquième scan :*

**Pour** chaque  $\mathbf{x}$  de  $\mathcal{X}$ ,  $\mathbf{y}$  de  $\{(x_1 - 1, x_2), (x_1, x_2 - 1)\}$   
**Si**  $testSegment(\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y}), \delta)$  Alors  
     **Si**  $\mathcal{D}(\mathbf{x}) \geq \mathcal{D}(\mathbf{y})$  Alors  $DMA(\mathcal{X}) = DMA(\mathcal{X}) \cup \mathbf{x}$   
     **Si**  $\mathcal{D}(\mathbf{x}) \leq \mathcal{D}(\mathbf{y})$  Alors  $DMA(\mathcal{X}) = DMA(\mathcal{X}) \cup \mathbf{y}$   
**Fin si**  
**Fin pour**

Procédure  $testSegment(\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y}), \delta)$

**Entrées :** Points extrémités du segment étudié;  
 Paramètre d'élagage  $\delta$   
**Sortie :** Vrai si  $\exists \mathbf{z} \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})] \mid \mathcal{D}(\mathbf{z}) > \delta$ , Faux sinon

$L = d(\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y}))$

**Si**  $L \leq 2\delta$  Alors renvoyer *Faux*

$\alpha^- = \alpha^+ = 0.5$

**Faire**

$z^- = \alpha^- \mathcal{P}(\mathbf{x}) + (1 - \alpha^-) \mathcal{P}(\mathbf{y})$ ,  $z^+ = \alpha^+ \mathcal{P}(\mathbf{x}) + (1 - \alpha^+) \mathcal{P}(\mathbf{y})$

**Si**  $\mathcal{D}(z^-) > \delta$  ou  $\mathcal{D}(z^+) > \delta$  Alors renvoyer *Vrai*

$\alpha^- = \alpha^- - \frac{\delta - \mathcal{D}(z^-)}{L}$

$\alpha^+ = \alpha^+ + \frac{\delta - \mathcal{D}(z^+)}{L}$

**Tant que**  $d(z^-, \mathcal{P}(\mathbf{x})) > \delta$  ou  $d(z^+, \mathcal{P}(\mathbf{y})) > \delta$

Renvoyer *Faux*

$$\begin{aligned} \delta_{max}(\mathbf{x}) &= \max_{\mathbf{z} \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})], \mathbf{y} \in V_4(\mathbf{x})} \mathcal{D}(\mathbf{z}) \\ \gamma_{max}(\mathbf{x}) &= \max_{\mathbf{y} \in V_4(\mathbf{x})} d(\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y})) \\ \lambda_{max}(\mathbf{x}) &= d(c, \mathcal{P}(\mathbf{x})) \text{ avec } c \text{ t.q. } d(c, \mathcal{P}(\mathbf{x})) = d(c, \mathcal{P}(\mathbf{y})) \forall \mathbf{y} \in V_4(\mathbf{x}) \end{aligned} \quad (3.10)$$

Plus cette valeur est élevée, plus le point  $\mathbf{x}$  sera résistant au processus d'élagage. Pour obtenir un résultat parfait pour une forme donnée, il est donc nécessaire de trouver une valeur du paramètre qui permette 1) que tous les points de l'axe médian attendu ait un paramètre maximal supérieur, et 2) que tous les autres points aient un paramètre maximal inférieur. Dans la pratique, cette séparation est difficile (voire impossible) à établir, ce qui explique l'importance et la difficulté du choix de la valeur du seuil.

Pour un panel d'images, et pour chacune des méthodes, ces valeurs sont normalisées et présentées sous la forme d'un dégradé de couleur Figure.3.9. On constate que les deux méthodes de l'état de l'art, à savoir GIMA et DLMA, présentent des résul-

tats semblables, qui sont dans l'ensemble clairement moins bons que ceux proposés par le DMA.

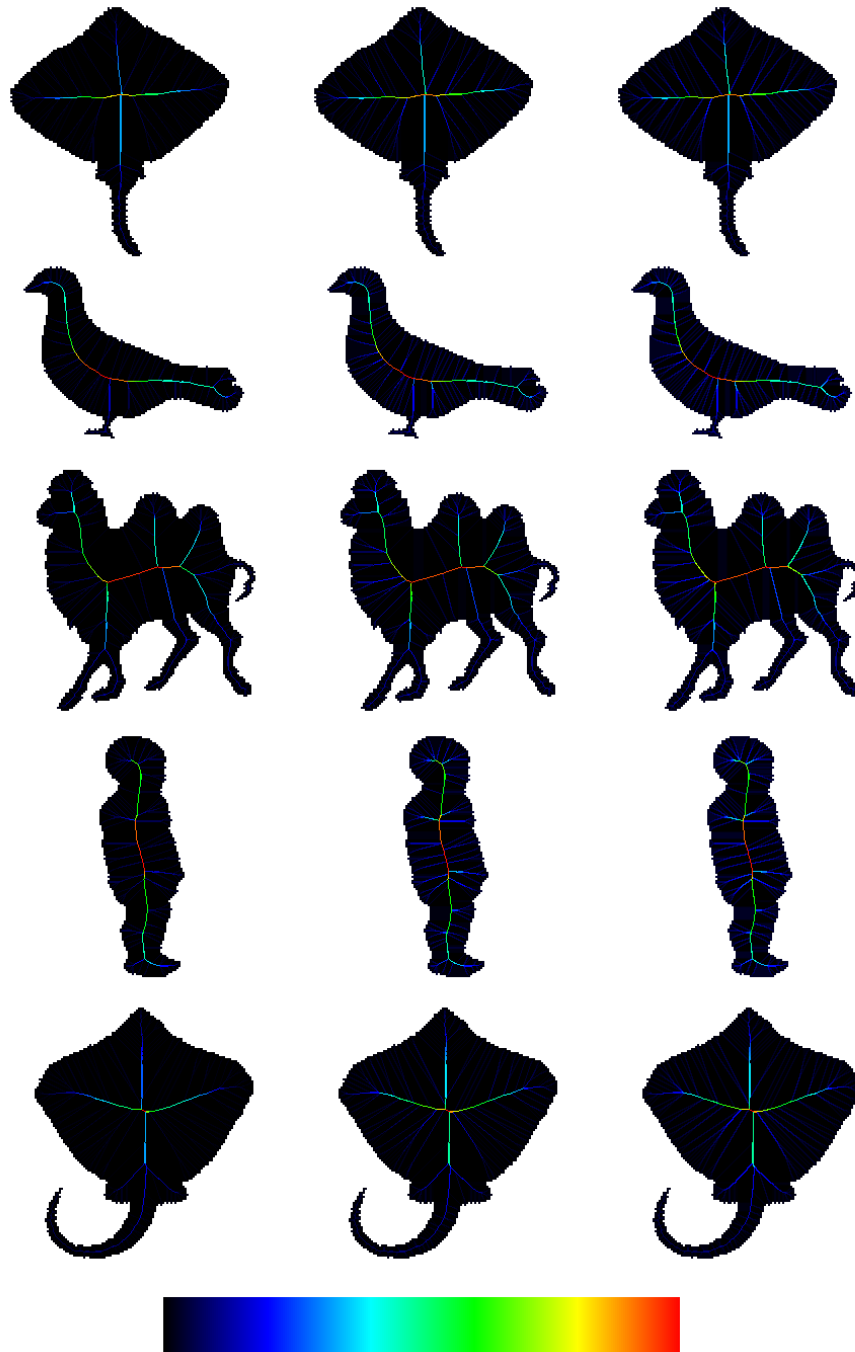
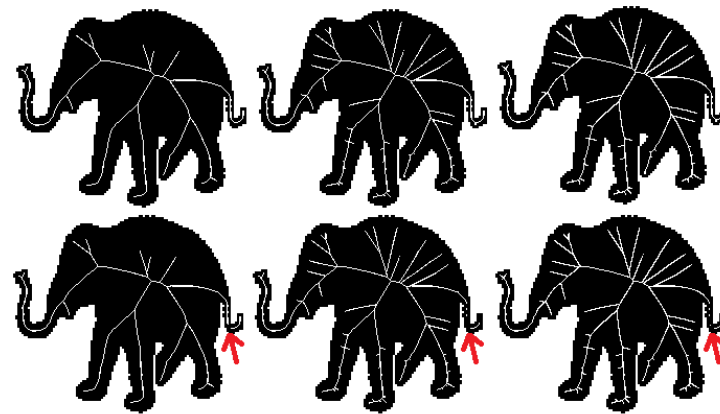
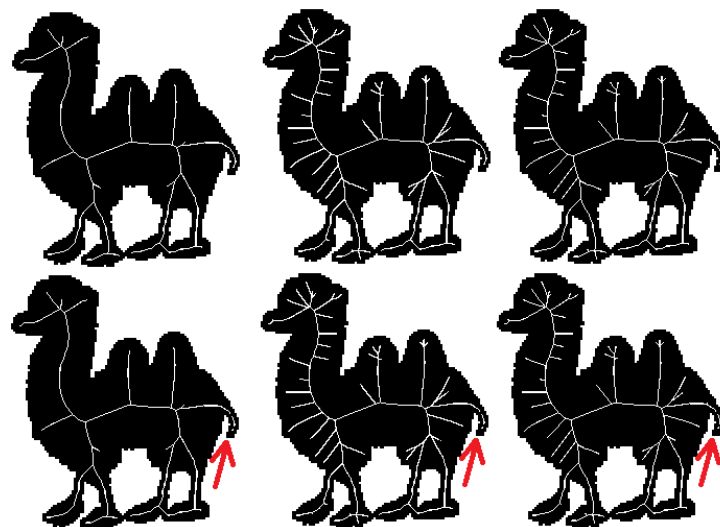


FIGURE 3.9 – Représentation des paramètres maximaux normalisés pour chacune des trois méthodes. De gauche à droite : DMA [Marie 2013c] GIMA [Hesselink 2007] et DLMA [Chaussard 2011].

Nous illustrons un peu plus ces propos en considérant deux exemples d'images (éléphant et chameau), pour lesquelles le paramètre de chacune des méthodes a été ajusté de manière optimale. A savoir, sa valeur est choisie comme étant la plus élevée qui permette de préserver les détails marquants de la forme : la queue et la trompe de l'éléphant et la queue du chameau. Les résultats obtenus sont présentés figure 3.10. Clairement, le nombre de branches parasites sur le DMA est largement inférieur à ce qui est observable pour les deux autres méthodes.



(a)



(b)

FIGURE 3.10 – Meilleurs squelettes homotopiques obtenus pour, de gauche à droite, le DMA [Marie 2013c], le GIMA [Hesselink 2007] et le DLMA [Chaussard 2011]. Pour chaque figure, la première ligne présente le meilleur résultat pour chacune des méthodes, tandis que la seconde met en évidence les zones où la contrainte d'homotopie est rompue (la connectivité du squelette est perdue).



### 3.4.4.3 Robustesse face aux déformations

Dans un second temps, nous avons cherché à évaluer la capacité du DMA à remplir l'un des objectifs majeurs fixés lors de l'élaboration de la méthode, à savoir sa capacité à produire un axe médian stable lorsqu'il prend en entrée une forme fixe sujette à un bruit de segmentation variable. Nous utilisons pour cela la série d'images de tombes présente dans la banque d'images. La figure 3.11 présente les résultats obtenus pour chaque méthode avec une valeur du paramètre fixe pour l'ensemble des images. Elle est choisie de telle sorte que la croix en reste une tout en réduisant au maximum les branches parasites. Outre une comparaison avec les méthodes de l'état de l'art qui est encore une fois à l'avantage du DMA, cette figure met en évidence le fait que l'erreur de segmentation doit être d'amplitude inférieure aux détails de la forme que l'on souhaite considérer, faute de quoi une branche non désirée est inévitable.

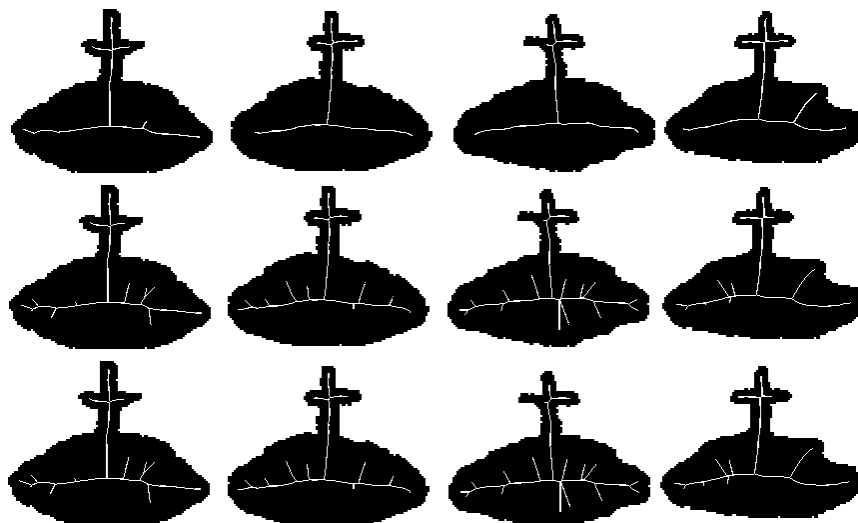


FIGURE 3.11 – Evolution des squelettes homotopiques obtenus lorsque la forme subit d'importantes déformations. En haut : DMA. Au centre : GIMA. En bas : DLMA

### 3.4.4.4 Complexité et temps de calcul

Le dernier critère de comparaison sur lequel nous nous sommes penchés est la question des performances computationnelles. L'un des objectifs fixés lors de l'élaboration du DMA était en effet de produire un algorithme suffisamment rapide et optimisé pour être exécuté en ligne directement par le robot. Le détail des solutions algorithmiques proposé dans la section précédente laisse entrevoir un comportement linéaire que nous avons voulu vérifier. Chacune des 216 images de la base Kimia [Sebastian 2001] est donc redimensionnée plusieurs fois (entre  $100 \times 100$  et  $1500 \times 1500$ ) avant que chacun des trois algorithmes y soit appliqué. La figure 3.12 présente la moyenne des temps de calcul observés pour l'ensemble des 216 images. A noter que le

processus de calcul des projections et distances au bord est évalué indépendamment puisqu'il est commun aux trois algorithmes.

Un premier constat est le comportement linéaire de chacune des trois méthodes. Ce n'est pas une surprise pour le GIMA et le DLMA, mais c'est une confirmation de l'intérêt des optimisations présentées dans la section précédente en ce qui concerne le DMA. L'indication principale de ces résultats reste néanmoins le bon comportement de notre approche relativement aux autres méthodes. Clairement, le GIMA est de loin l'algorithme le plus rapide, puisque pour une résolution de  $800 \times 600$  (c'est-à-dire la résolution des images acquises par le robot), il ne lui faut que 9.4ms en moyenne pour s'exécuter (contre 27.3 pour le DLMA). Le DMA nécessite quant à lui 10.1ms, ce qui, relativement à la complexité du critère d'élagage, est tout à fait acceptable. Quoi qu'il en soit, en ajoutant le temps nécessaire au calcul des projections et distances au bord, ces trois méthodes restent adaptées à une application temps réel, puisqu'elles offrent des performances supérieures à 20Hz.

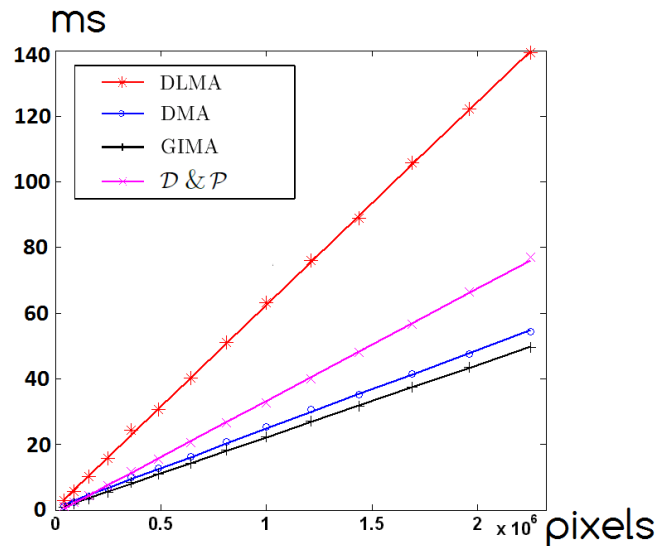


FIGURE 3.12 – Temps de calcul en fonction du nombre de pixels pour chacune des trois méthodes comparées. Le DLMA [Chaussard 2011] est clairement le plus lent tandis que contrairement à ce que peut laisser penser la formulation dans chacun des cas, le DMA offre des performances comparables à celles du GIMA [Hesselink 2007].

### 3.4.5 Première application : cartographie topologique d'une vue aérienne

Dans le cadre du projet ANR R-Discover, une carte aérienne est utilisée en entrée d'algorithmes [Lozenguez 2011, Lozenguez 2012] de répartition de zones à une cohorte de robots, dans un but de couverture de l'espace. Initialement, cette carte était construite à la main. L'idée est donc venue d'automatiser le processus en utilisant le DMA sur une vue aérienne du scénario, où les voies navigables sont

segmentées (voir Fig. 3.13). Un traitement est ensuite réalisé afin d'une part d'y détecter les noeuds topologiques, et d'autre part de fournir une notion de longueur à chaque arête. Cette section présente ce travail.

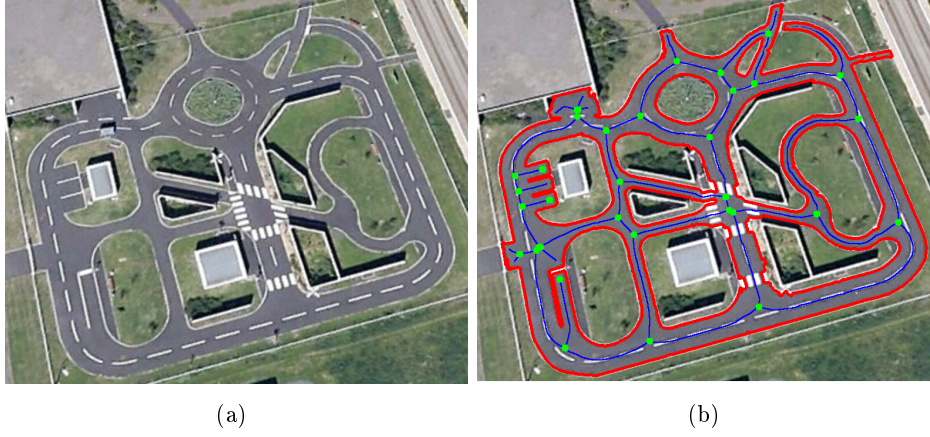


FIGURE 3.13 – (a) Vue aérienne du site de Pavin (Clermont-Ferrand) obtenue à partir de Google Earth. (b) Graphe topologique obtenu après segmentation de l'espace navigable et application du DMA.

### 3.4.5.1 Vers le squelette de l'environnement

Soit  $FS_w \cup \overline{FS_w}$  une vue d'un environnement se présentant sous la forme d'un masque binaire (voir Figure 3.14(a)), où  $FS_w$  désigne l'espace navigable pour le robot. Chaque voie navigable peut être associée à l'une des branches du squelette  $AM(FS_w)$  de  $FS_w$ . Elles sont donc extraites par application du DMA, en choisissant une valeur de  $\delta$  qui corresponde au rayon présumé d'un cercle englobant la projection d'un robot dans cette vue.

Le résultat obtenu est un nouveau masque binaire (voir Fig.3.14(b)). Il doit ensuite être transformé en graphe topologique, c'est-à-dire en une collection de noeuds marquant les intersections de l'environnement, liés par des relations d'adjacence qui correspondent aux différentes voies navigables.

### 3.4.5.2 Recherche des noeuds topologiques

Dans notre cas, un noeud topologique est un point du DMA qui se situe à l'intersection de plusieurs branches du squelette obtenu. On le définit comme suit :

**Définition 2** Soit  $\delta \in \mathbb{R}_+^*$ .  $\mathbf{x}$  est un noeud topologique au sens du  $\delta$ -medial axis s'il vérifie les propriétés suivantes :

$$\exists \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{V}_8(\mathbf{x}) \text{ t.q. } \begin{cases} \exists \mathbf{z}_1 \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y}_1)] \mid \mathcal{D}(\mathbf{z}_1) \geq \delta \\ \exists \mathbf{z}_2 \in [\mathcal{P}(\mathbf{x}), \mathcal{P}(\mathbf{y}_2)] \mid \mathcal{D}(\mathbf{z}_2) \geq \delta \\ \exists \mathbf{z}_3 \in [\mathcal{P}(\mathbf{y}_1), \mathcal{P}(\mathbf{y}_2)] \mid \mathcal{D}(\mathbf{z}_3) \geq \delta \\ \mathcal{D}(\mathbf{x}) \geq \mathcal{D}(\mathbf{y}_1) \ \& \ \mathcal{D}(\mathbf{x}) \geq \mathcal{D}(\mathbf{y}_2) \end{cases} \quad (3.11)$$

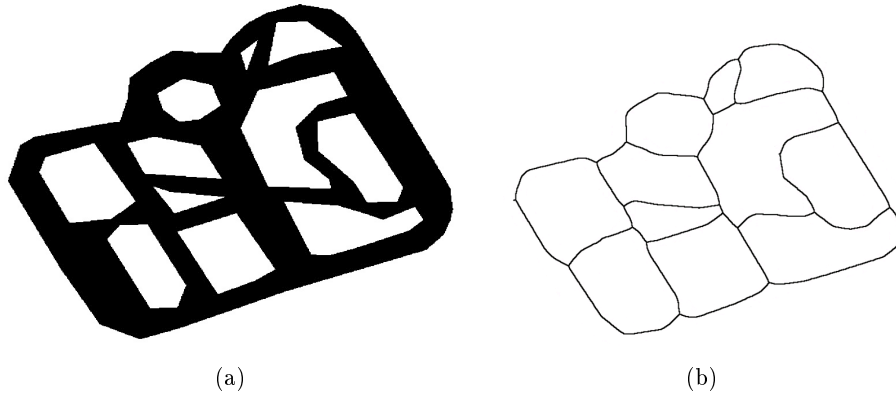


FIGURE 3.14 – (a) Masque binaire délimitant les frontières de l'espace navigable dans la vue aérienne. (b) Voies navigables extraites à partir du delta medial axis.

Autrement dit,  $\mathbf{x}$  est un noeud topologique, si avec au moins deux de ses voisins il vérifie les conditions requises pour appartenir au DMA, et si en plus lesdits voisins remplissent les mêmes conditions. La figure.3.15(a) schématise les différentes notations utilisées dans un cas où  $\mathbf{x}$  est bien un noeud topologique tandis que la figure.3.15(b) illustre la nécessité de la troisième condition.

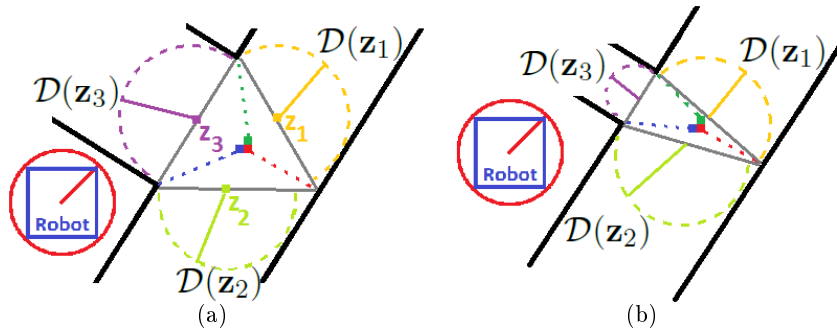


FIGURE 3.15 – (a) Cas où le point rouge **est** un noeud topologique : clairement,  $\mathcal{D}(z_1)$ ,  $\mathcal{D}(z_2)$ , et  $\mathcal{D}(z_3)$  sont supérieures à  $\delta$ . (b) Cas où le point rouge **n'est pas** un noeud topologique au sens de  $\delta$  :  $\mathcal{D}(z_3) < \delta$ . A noter que  $\delta$  est choisi afin de former un cercle en adéquation avec les dimensions du robot.

D'un point de vue algorithmique, chaque point  $\mathbf{x}$  de  $\text{DMA}(FS_w)$  est associé à un accumulateur qui mémorise les points de son voisinage avec lesquels il vérifie le test du DMA. Si l'un des couples qu'il est possible de former au sein de cet accumulateur permet de remplir les mêmes conditions, alors  $\mathbf{x}$  vérifie l'équation 3.11, et peut être ajouté à la liste des noeuds. Afin d'éviter que plusieurs points désignent le même noeud topologique, nous ajoutons la condition qu'aucun de ses voisins ne le soit déjà.

### 3.4.5.3 Génération des relations d'adjacence

Reste à définir les relations d'adjacence qui lient les noeuds extraits. L'idée est pour cela de parcourir les branches du squelette en simulant le déplacement d'un ensemble de robots dans l'image, chacun étant chargé de l'exploration d'une branche. Soient  $p_t^r$  la position (dans l'image) d'un robot  $r$  à l'instant  $t$  et  $N^r$  le noeud duquel il est parti. Les déplacements des robots sont régis par l'intersection de l'axe médian extrait avec un cercle de rayon  $\delta$  dont ils sont le centre. Plusieurs cas de figures sont susceptibles de survenir :

- Une intersection : la branche explorée mène à une impasse, l'intersection n'est rien d'autre que  $p_{t-1}^r$ . Aucune relation d'adjacence entre noeud n'est créée, le "robot" est supprimé de la liste.
- Deux intersections : le robot est au coeur de la branche et poursuit son exploration. L'intersection qui correspond à  $p_{t-1}^r$  est ignorée. L'autre intersection sert de position  $p_{t+1}^r$  pour l'itération suivante.
- Trois intersections ou plus : le robot est proche d'un noeud topologique. La distance entre  $p_t^r$  et chaque noeud est évaluée, et la plus petite permet de définir le noeud  $N_i$  atteint. Une relation d'adjacence est alors créée entre les noeuds  $N^r$  et  $N_i$ . Le robot  $r$  est ensuite supprimé, et pour chaque branche non explorée partant de  $N_i$ , un nouveau robot est initialisé.

L'algorithme s'arrête lorsque la liste des robots en mouvement est vide et que tous les noeuds connus ont été visités (voir Figure 3.16 pour une illustration).

### 3.4.5.4 Evaluation de la longueur des arcs

La matrice d'adjacence ainsi obtenue contient une information binaire pour chaque couple de noeuds (existence d'une liaison ou non). Dans un cadre applicatif comme celui de la robotique mobile, il peut être intéressant de posséder une information de distance entre noeuds. Nous nous plaçons à un instant  $t$  où un robot  $r$  atteint un noeud  $N_i$ . La distance entre le noeud  $N^r$  sur lequel il a été initialisé, et le noeud  $N_i$  qu'il vient d'atteindre peut être approximée à partir du nombre d'itérations  $n$  nécessaires au robot pour atteindre  $N_i$ . La distance  $d(N^r, N_i)$  est alors définie par :

$$d(N^r, N_i) = n\delta + d(p_t^r, N_i) \quad (3.12)$$

Le résultat final est présenté Figure 3.17 sous forme d'image. Dans la pratique, seule la matrice d'adjacence contenant les distances entre noeuds est utilisée en entrée des algorithmes d'exploration [Lozenguez 2011, Lozenguez 2012].

## 3.5 Adaptation aux images omnidirectionnelles

### 3.5.1 Distortions et axe médian

L'algorithme du *Delta Medial Axis* est calculé directement dans l'image, et ne tient pas compte du contexte lié à sa formation. Autrement dit, il permet de calculer

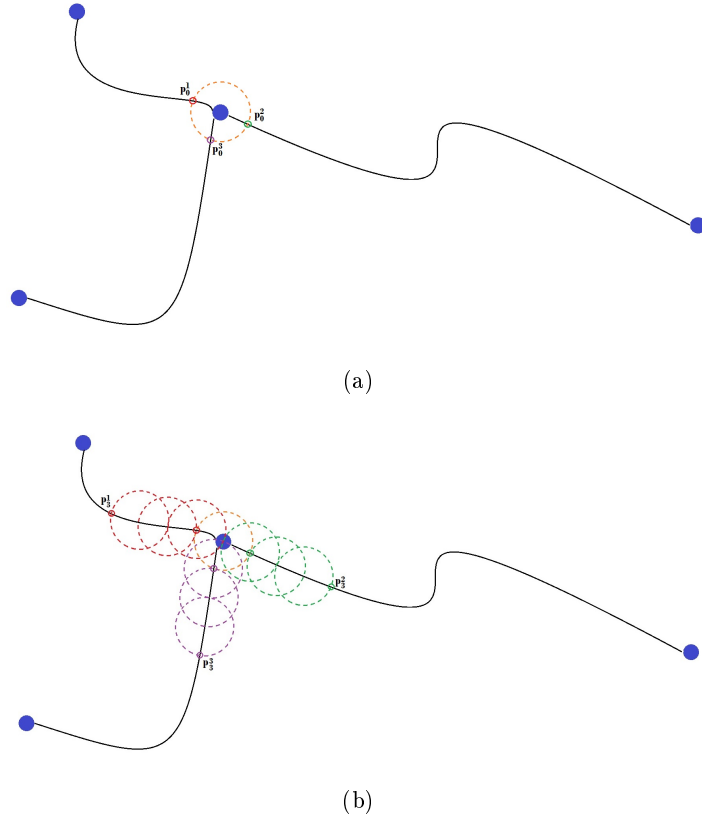


FIGURE 3.16 – Illustration du processus d’exploration des branches du squelette afin de définir les relations d’adjacence entre noeuds. (a) initialisation de trois robots virtuels autour du noeud central. (b) Position des robots après 4 itérations.

le squelette de la projection de l’objet étudié au sens de la métrique considérée, qui ne correspond que très rarement à la projection du squelette de ce même objet. C’est particulièrement vrai dans notre cas où la géométrie du capteur implique de très fortes distorsions lors du processus de projection. Pourtant, l’objectif visé est de caractériser la topologie de la scène, et non celle de sa projection dans l’image. Il est donc temps de considérer en détail les particularités du cadre applicatif qui est le nôtre afin de proposer une version du DMA qui y soit adaptée. Elle doit être pour cela capable de tenir compte du processus de formation de l’image, afin d’y extraire directement la projection du squelette de l’espace navigable.

Soient  $FS_W$  l’espace libre 3D (c’est-à-dire le sol, supposé plan, sur lequel le robot peut se déplacer sans risque) et  $AM(FS_W)$  le squelette qui en découle. Nous cherchons à construire une version adaptée du DMA, notée ODMA (pour *Omnidirectional  $\delta$ -medial axis*) telle que :

$$ODMA(FS) = f(AM(FS_W)) \quad (3.13)$$

où, rappelons le,  $f$  correspond à la fonction de projection entre la scène et le plan

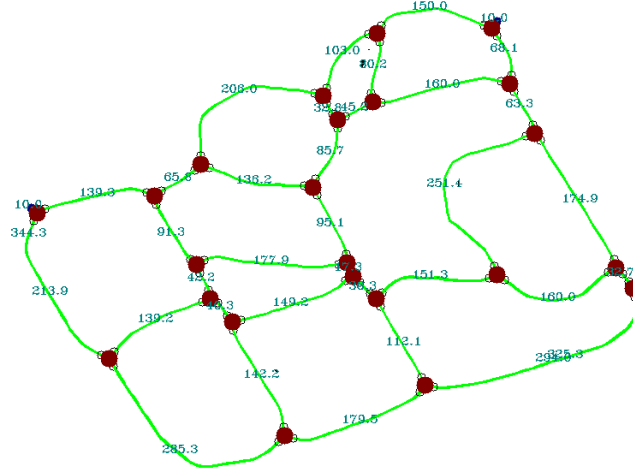


FIGURE 3.17 – Représentation graphique du graphe topologique extrait à partir des algorithmes présentés.

image. Cela revient à définir dans l'image une nouvelle métrique  $\tilde{d}$  conforme à la distance euclidienne dans le plan du sol (définie à un facteur d'échelle près). Soient  $\mathbf{P}_1$  et  $\mathbf{P}_2$  deux points du sol, et  $\mathbf{p}_1$  et  $\mathbf{p}_2$  leur projection dans l'image. On a :

$$\tilde{d}(\mathbf{p}_1, \mathbf{p}_2) = \alpha d(\mathbf{P}_1, \mathbf{P}_2) = \alpha d(f^{-1}(\mathbf{p}_1), f^{-1}(\mathbf{p}_2)) \quad (3.14)$$

avec  $\alpha$  un facteur d'échelle.

L'évaluation d'une telle métrique passe par l'inversion de la fonction  $f$ , donc des équations de projection présentées dans le chapitre précédent.

### 3.5.2 Projection inverse

Commençons par rappeler que  $f = f_{img} \circ f_{sph}$  est le résultat de la composition de deux fonctions de projections intermédiaires qui s'écrivent :

$$f_{sph} : \mathbf{P} \in \mathbb{R}^3 \mapsto \mathbf{P}_s \in \mathcal{S}^2; t.q. \begin{cases} X_s = X/\rho \\ Y_s = Y/\rho \\ Z_s = Z/\rho \end{cases}, \quad \rho = \sqrt{X^2 + Y^2 + Z^2} \quad (3.15)$$

$$f_{img} : \mathbf{P}_s \in \mathcal{S}^2 \mapsto \mathbf{p} \in \mathbb{R}^2; t.q. \begin{cases} x = X_s/(Z_s + \xi) \\ y = Y_s/(Z_s + \xi) \end{cases} \quad (3.16)$$

L'inverse de cette fonction, notée  $f^{-1}$  peut s'exprimer par l'intermédiaire des lois de composition  $f^{-1} = f_{sph}^{-1} \circ f_{img}^{-1}$ . Comme vu au chapitre précédent, l'inversion de  $f_{img}$  ne pose aucun problème, et on trouve :

$$\mathbf{P}_s = f_{img}^{-1}(\mathbf{p}) : \begin{cases} X_s = A(\mathbf{p})x \\ Y_s = A(\mathbf{p})y \\ Z_s = A(\mathbf{p}) - \xi \end{cases} \quad \text{avec } A(\mathbf{p}) = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} \quad (3.17)$$

Pour l'opérateur  $f_{sph}$ , nous faisons l'hypothèse que la caméra est positionnée de telle sorte que le plan du sol soit parallèle au plan image. Nous partons également du principe que les points projetés appartiennent tous au plan du sol, et partagent donc la même altitude ( $Z = H$ ).  $f_{sph}$  peut alors être inversée, et on trouve :

$$\mathbf{P} = f_{sph}^{-1}(\mathbf{P}_s) : \begin{cases} X = H X_s / Z_s \\ Y = H Y_s / Z_s \\ Z = H \end{cases} \quad (3.18)$$

En combinant les équations 3.17 et 3.18, l'opérateur de projection inverse s'écrit :

$$\mathbf{P} = f^{-1}(\mathbf{p}) : \begin{cases} X = H A(\mathbf{p})x / (A(\mathbf{p}) - \xi) \\ Y = H A(\mathbf{p})y / (A(\mathbf{p}) - \xi) \\ Z = H \end{cases} \quad (3.19)$$

### 3.5.3 Vers une métrique adaptée

A partir de (3.14), deux points du sol  $\mathbf{P}_1 = (X_1, Y_1, H)^t$  et  $\mathbf{P}_2 = (X_2, Y_2, H)^t$  sont liés par la relation :

$$\tilde{d}(\mathbf{P}_1, \mathbf{P}_2)^2 = \alpha^2((X_1 - X_2)^2 + (Y_1 - Y_2)^2) \quad (3.20)$$

Qui, en posant  $\alpha = 1/H^2$  et en utilisant (3.19), mène à l'expression finale de notre métrique adaptée :

$$\tilde{d}(\mathbf{P}_1, \mathbf{P}_2)^2 = \frac{A^2(\mathbf{p}_1)}{(A(\mathbf{p}_1) - \xi)^2} (x_1^2 + y_1^2) + \frac{A^2(\mathbf{p}_2)}{(A(\mathbf{p}_2) - \xi)^2} (x_2^2 + y_2^2) - 2 \frac{A(\mathbf{p}_1)A(\mathbf{p}_2)}{(A(\mathbf{p}_1) - \xi)(A(\mathbf{p}_2) - \xi)} (x_1 x_2 + y_1 y_2) \quad (3.21)$$

avec  $A(\mathbf{p}_i) = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x_i^2 + y_i^2)}}{x_i^2 + y_i^2 + 1}$ ,  $i = 1, 2$

### 3.5.4 Formalisation du ODMA

La métrique (3.21) ainsi définie permet de générer dans l'image une projection et une distance au bord adaptées qui s'expriment par :

$$\forall \mathbf{x} \in FS, \begin{cases} \tilde{\mathcal{D}}(\mathbf{x}, \partial FS) = \min_{\mathbf{y} \in \partial FS} \tilde{d}(\mathbf{x}, \mathbf{y}) \\ \tilde{\mathcal{P}}(\mathbf{x}, \partial FS) = \left\{ \mathbf{y} \in \partial FS \mid \tilde{d}(\mathbf{x}, \mathbf{y}) = \tilde{\mathcal{D}}(\mathbf{x}) \right\} \end{cases} \quad (3.22)$$

La différence avec la distance au bord et la projection fondées sur une métrique euclidienne dans l'image est illustrée Figure 3.18.

A partir de ces versions adaptées, le ODMA peut finalement être défini :



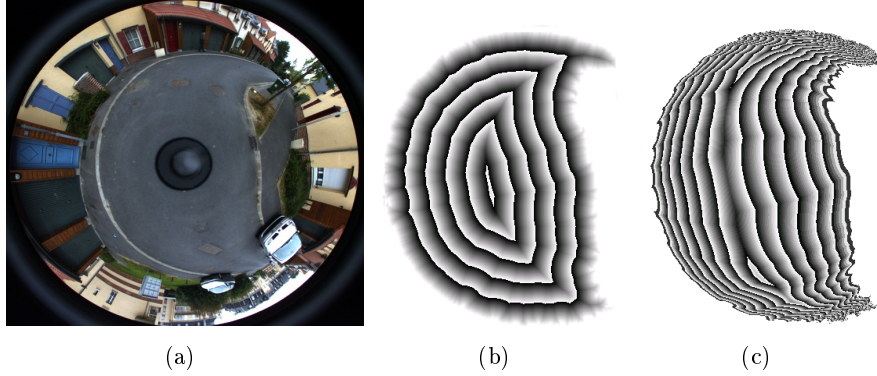


FIGURE 3.18 – (a) Image omnidirectionnelle acquise par le robot. (b) Courbes de niveau des distances au bord pour une métrique euclidienne. (c) Courbes de niveau des distances au bord obtenues avec la métrique adaptée.

**Définition 3** Soit  $\tilde{\delta} \in \mathbb{R}_+^*$  et  $FS$  l'espace navigable supposé plan observé depuis le capteur catadioptrique. Le omnidirectional  $\tilde{\delta}$ -medial axis (ODMA) de  $FS$ , est l'ensemble des points  $\mathbf{x} \in FS$  qui vérifient les propriétés suivantes :

$$\exists \mathbf{y} \in FS, \|\mathbf{x} - \mathbf{y}\| = 1 \text{ t.q. } \begin{cases} \exists \mathbf{z} \in [\tilde{\mathcal{P}}(\mathbf{x}), \tilde{\mathcal{P}}(\mathbf{y})] \mid \tilde{\mathcal{D}}(\mathbf{z}) \geq \tilde{\delta} \\ \tilde{\mathcal{D}}(\mathbf{x}) \geq \tilde{\mathcal{D}}(\mathbf{y}) \end{cases} \quad (3.23)$$

A noter que cette fois les points sont exprimés dans le plan image normalisé, et non plus dans le repère pixelique. La fonction qui permet de passer de l'un à l'autre est fondée sur la matrice des paramètres intrinsèques et nécessite donc que le capteur soit calibré (c'est de toute façon nécessaire dans le processus de construction de signature).

### 3.5.5 Réglage automatique de $\tilde{\delta}$

A la différence de la version standard du DMA, qui a pour vocation d'être la plus générale possible, la version adaptée est spécifiquement conçue pour répondre aux particularités de notre cadre applicatif. Notamment, l'évaluation de la métrique adaptée sous-entend une maîtrise de la position de la caméra et un postulat sur la nature planaire du sol. Rappelons que l'objectif est de fournir au robot un outil lui permettant de se déplacer sans risque. Le squelette extrait doit donc représenter exclusivement des trajectoires sûres sur lesquelles baser le processus de navigation. Pourtant, la scène observée ne tient pas compte de la géométrie du robot. Dans certaines situations, il peut arriver que des portions de l'espace libre soient trop étroites pour lui permettre de les traverser. Nous proposons donc de choisir la valeur de  $\tilde{\delta}$  en fonction des dimensions du robot.

La condition (3.23) stipule que deux points de la frontière généreront une branche de squelette si et seulement si à une quelconque position le long du segment les reliant, il existe un point  $\mathbf{z}$  dont la distance au bord adaptée  $\tilde{\mathcal{D}}(\mathbf{z})$  est supérieure

à  $\tilde{\delta}$ . En considérant le robot comme une forme circulaire ayant ce rayon, les seules branches générées correspondront à des trajectoires qui non seulement sont sûres au sens du robot (c'est-à-dire que ses dimensions lui permettent de s'y déplacer sans danger), mais qui en plus mènent à des particularités topologiques intéressantes à explorer (typiquement des voies navigables), puisqu'il est garanti que le robot puisse au moins se placer entre les deux points du bord les ayant générées.

Nous choisissons la valeur de  $\tilde{\delta}$  en nous inspirant de l'initialisation du contour actif décrite dans le chapitre précédent. Autrement dit :

$$\tilde{\delta} = \tilde{d}(c, p) \quad (3.24)$$

où  $p$  est un point appartenant au plus petit cercle encapsulant la projection du robot dans l'image, et  $c$  est le centre optique de la caméra.

### 3.5.6 Quid de l'implémentation ?

Pour rappel, la version standard du DMA procède en deux étapes. Dans un premier temps, l'image est scannée afin de calculer pour chaque pixel la projection et la distance au bord. Une seconde étape permet, en utilisant ces deux outils, de déterminer les points qui forment le DMA.

Pour calculer le ODMA, plusieurs modifications sont nécessaires. Premièrement, l'algorithme de Hirata [Hirata 1996] qui permet de calculer la projection et la distance au bord en chaque point nécessite l'utilisation d'une métrique basée sur une norme axiale symétrique, ce qui n'est pas le cas de  $\tilde{d}$ . Une alternative doit donc être trouvée, en gardant à l'esprit que l'une des priorités est de conserver une complexité linéaire. Le choix s'est porté sur le 8-SED (8-neighborhood Squared Euclidian Distance) proposé par Danielsson [Danielsson 1980], qui possède cette propriété, mais n'est capable de fournir qu'une approximation de la distance au bord en chaque point.

Cette approche est fondée sur le postulat que chaque point de l'image partage sa projection avec au moins l'un de ses 8 voisins. En réalité, ce n'est pas toujours le cas (voir Figure. 3.19), ce qui constitue la source de l'approximation (l'erreur générée est largement discutée et quantifiée dans [Danielsson 1980]). En deux parcours complets de l'image (un dans chaque sens), les projections se propagent de pixel en pixel, la distance au bord pouvant alors en être trivialement déduite. A la différence de l'algorithme de Hirata, rien n'empêche ici d'utiliser une autre métrique que la distance euclidienne. En particulier, avec notre métrique adaptée, le postulat de base (un point partage sa projection avec un de ses voisins) reste valide. De là, l'adaptation de l'algorithme (présentée Algo 5) ne pose pas de difficulté particulière.

La seconde étape (labellisation des pixels appartenant au ODMA) ne nécessite pas de changement majeur. Chaque fois que nécessaire, la métrique adaptée est utilisée en lieu et place de la distance euclidienne. En revanche, l'algorithme en temps que tel n'offre pas des performances temps réel. A chaque étape, un nombre considérable de mesures de distances nécessite en effet d'être réalisé. Etant donné l'expression de la métrique adaptée (voir Eq.3.21), le temps requis pour l'évaluation

**Algorithme 5** Calcul des distances au bord et projections adaptées

**Entrée :** Masque binaire  $\mathbf{FS} \cup \overline{\mathbf{FS}}$  de dimensions  $m * n$

**Sorties :**  $\tilde{\mathcal{P}}(\mathbf{x})$  et  $\tilde{\mathcal{D}}(\mathbf{x})$  pour chaque  $\mathbf{x} \in \mathbf{FS} \cup \overline{\mathbf{FS}}$

*Initialisation :*

**Pour** chaque  $\mathbf{x}$  de  $\mathbf{FS}$ ,  $\tilde{\mathcal{D}}(\mathbf{x}) = \infty$ ,  $\tilde{\mathcal{P}}(\mathbf{x}) = \emptyset$

**Pour** chaque  $\mathbf{x}$  de  $\overline{\mathbf{FS}}$ ,  $\tilde{\mathcal{D}}(\mathbf{x}) = 0$ ,  $\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{x}$

*Premier et second scans :*

**Pour**  $i = 2$  à  $m$

**Pour**  $j = 2$  à  $n$

*testPixel*(( $i, j$ ),  $mask_1$ )

**Pour**  $j = n - 1$  à  $1$

*testPixel*(( $i, j$ ),  $mask_2$ )

*Troisième et quatrième scans :*

**Pour**  $i = m - 1$  à  $1$

**Pour**  $j = 2$  à  $n$

*testPixel*(( $i, j$ ),  $mask_3$ )

**Pour**  $j = n - 1$  à  $1$

*testPixel*(( $i, j$ ),  $mask_4$ )

**Procédure testPixel(x,mask)**

**Entrées :** Point  $\mathbf{x}$  étudié et masque  $mask$  des voisins considérés (voir Fig. 3.20)

**Sortie :**  $\emptyset$ , les modifications sont appliquées à  $\mathbf{x}$  directement dans la procédure

**Si**  $\mathbf{x} \in \mathbf{FS}$  Alors

**Pour** chaque  $e \in mask$

$\mathbf{y} = \mathbf{x} + e$

**Si**  $\tilde{\mathcal{D}}(\mathbf{x}) > \tilde{d}(\mathbf{x}, \tilde{\mathcal{P}}(\mathbf{y}))$  Alors  $\tilde{\mathcal{P}}(\mathbf{x}) = \tilde{\mathcal{P}}(\mathbf{y})$ ,  $\tilde{\mathcal{D}}(\mathbf{x}) = \tilde{d}(\mathbf{x}, \tilde{\mathcal{P}}(\mathbf{y}))$

**Fin** pour

**Fin** si

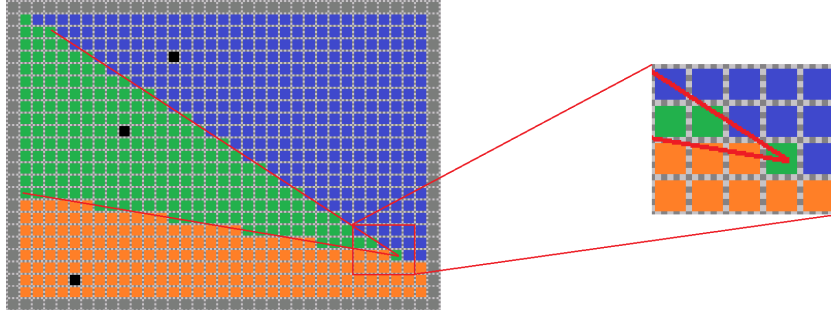


FIGURE 3.19 – Illustration de l'erreur générée par l'algorithme de Danielsson. En noir, les 3 points frontières. En vert, orange et bleu, leurs régions de Voronoi respectives. L'agrandissement à droite montre que le point au centre est plus proche du point frontière central, ce qui n'est le cas d'aucun de ses voisins directs. Il sera donc mal classifié par l'algorithme de Danielsson.

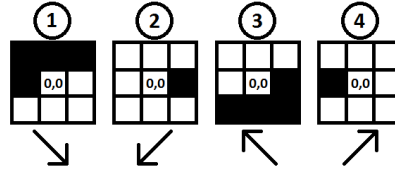


FIGURE 3.20 – Les masques utilisés dans l'algorithme de Danielsson (voir Algo.5)

de chacune d'entre elles est beaucoup trop élevé. Pour chaque pixel  $\mathbf{p} = (x, y)^t$  du plan image normalisé, on note :

$$\begin{aligned} B(\mathbf{p}) &= A(\mathbf{p}) / (A(\mathbf{p}) - \xi) \\ C(\mathbf{p}) &= B(\mathbf{p})^2 (x^2 + y^2) \end{aligned} \quad (3.25)$$

La métrique  $\tilde{d}$  peut alors être réécrite, et on obtient pour tout couple de pixels  $\mathbf{p}_1 = (x_1, y_1)^t$ ,  $\mathbf{p}_2 = (x_2, y_2)^t$  (toujours exprimés dans le plan image normalisé) :

$$\tilde{d}(\mathbf{p}_1, \mathbf{p}_2) = C(\mathbf{p}_1) + C(\mathbf{p}_2) - 2B(\mathbf{p}_1)B(\mathbf{p}_2)(x_1x_2 + y_1y_2) \quad (3.26)$$

On remarque qu'à la fois  $B$  et  $C$  peuvent être calculés une seule fois pour chaque pixel de l'image. Nous utilisons deux look-up tables (LUT) qui permettent de considérablement réduire le nombre d'opérations nécessaires à l'évaluation de la métrique adaptée (voir Tableau. 3.1).

### 3.5.7 Résultats

#### 3.5.7.1 Cadre expérimental

Le ODMA étant élaboré dans un cadre très spécifique, il n'existe pas de méthodes comparables dans la littérature. Les résultats présentés dans cette section ne cherchent donc pas à quantifier l'amélioration des résultats que permet la version

Opération/Opérateur	+	-	×	÷	√
$d(p_1, p_2)$	1	2	2	-	-
$A(p)$	5	1	6	1	1
$\tilde{d}(p_1, p_2)$	44	13	60	8	8
$\tilde{d}(p_1, p_2)$ (avec LUT)	2	1	5	-	-

TABLE 3.1 – Tableau présentant le nombre et le type des opérations requises pour le calcul de différentes fonctions.

adaptée, mais plutôt de justifier son utilité relativement à la version standard. La banque d'images utilisée est cette fois issue d'un scénario en environnement extérieur qui offre plusieurs topologies de l'espace libre intéressantes (carrefour, courbes, ...). Pour chaque image acquise, l'espace libre est extrait en utilisant l'algorithme présenté au chapitre précédent.

### 3.5.7.2 Corrélation avec le diagramme de Voronoï

Nous souhaitons dans un premier temps évaluer la capacité de chacune des deux versions de l'algorithme à remplir l'objectif fixé, à savoir extraire la topologie de la scène perçue en capturant la projection de son squelette. La figure 3.21 présente l'étude de quatre situations différentes. Pour chacune d'entre elles, cinq imagettes représentent dans l'ordre :

- l'image acquise par le robot, choisie pour son intérêt topologique
- L'axe médian obtenu avec la version standard du DMA (pour une valeur de  $\delta$  optimale)
- Une représentation  $2D$  du sol sur laquelle ont été reprojctées les frontières de l'espace navigable ainsi que le DMA extrait
- L'axe médian obtenu avec la version omnidirectionnelle ( $\tilde{\delta}$  est, on le rappelle, automatiquement déterminé)
- Une représentation  $2D$  du sol sur laquelle ont été reprojctées les frontières de l'espace navigable ainsi que le ODMA extrait

Comme les deuxième et quatrième lignes permettent de le constater, le squelette obtenu pour chacune des deux versions dénote une topologie cohérente avec l'espace libre extrait. En se basant uniquement sur ces résultats, on pourrait même considérer que la version standard offre des résultats de meilleure qualité, puisque des branches parasites ont tendance à apparaître sur le ODMA à des positions 3D éloignées du robot (la résolution diminuant de manière dramatique lorsque la distance augmente). Les reprojctions dans le plan du sol rendent toutefois un tout autre verdict. Clairement, la version standard produit un résultat qui n'est en aucun cas en adéquation avec le diagramme de Voronoï de la scène perçue. C'est un problème dans un contexte de navigation autonome, puisqu'en se basant sur ce résultat, le robot ne va jamais emprunter les routes les plus sûres (c'est particulièrement visible

sur la topologie en forme de courbe). Au contraire, lorsque la distance au robot n'est pas excessive, le ODMA propose une approximation très correcte du diagramme de Voronoï, et remplit donc les objectifs fixés. Le fait que la qualité du résultat se détériore lorsque la distance au robot augmente n'est en soit qu'un problème mineur et facile à solutionner : puisque le robot se déplace, la topologie est mise à jour à chaque nouvelle image, et les régions trop lointaines à l'instant  $t$  vont soit disparaître du champ de vue et ne présenter qu'un intérêt négligeable, soit s'approcher à l'instant  $t + 1$ , et par conséquent se trouver dans la zone de "confort" de l'algorithme, c'est-à-dire là où ses performances sont les meilleures.

### 3.5.7.3 Stabilité dans le temps

La topologie locale n'a pas pour seule vocation à être utilisée dans une optique de navigation autonome. Nous aurons notamment besoin de détecter les noeuds topologiques dans l'environnement (c'est-à-dire les positions 3D où plusieurs voies navigables s'intersectent). C'est impossible avec la version standard de l'algorithme, puisque la reprojexion sur le plan du sol d'un tel point se déplace lorsque le point de vue du robot change. Pour illustrer ce phénomène, la Figure.3.22 présente la superposition de trois topologies extraites à différentes positions d'un même carrefour. Clairement, pour le ODMA, les topologies sont pratiquement superposées, et le noeud topologique se résume à une région aux dimensions réduites. Pour la version standard en revanche, ce n'est absolument pas le cas.

## 3.6 Conclusion

Dans ce chapitre, la question de l'extraction de l'axe médian d'une forme 2D discrète a été abordée. Après une étude approfondie de l'outil, c'est-à-dire son histoire, ses propriétés, mais aussi et surtout les enjeux et difficultés qui y sont liés, nous avons proposé le *Delta Medial Axis*, un nouvel algorithme qui intègre un critère d'élagage original et dont le cadre applicatif va bien plus loin que le contexte robotique. Une implémentation de complexité quasi linéaire est proposée, et les résultats expérimentaux montrent une réelle amélioration par rapport aux solutions existantes. La deuxième partie du chapitre a quant à elle été dédiée à notre contexte en particulier, en proposant une version adapté du DMA capable de considérer l'anamorphose du système de projection catadioptrique afin de produire un squelette cohérent avec la scène observée. De même, en prenant en compte les dimensions du robot, une valeur de delta justifiée peut être déterminée, ce qui permet non seulement de rendre le processus totalement automatique, mais aussi et surtout d'éliminer d'éventuelles branches générées par des voies trop étroites pour permettre au robot de s'y déplacer. Au final, la représentation construite est en adéquation avec les objectifs fixés puisque le résultat correspond à la projection dans l'image du diagramme de Voronoï (filtré) de la scène perçue, et fournit donc un ensemble de trajectoires sûres pour le robot.

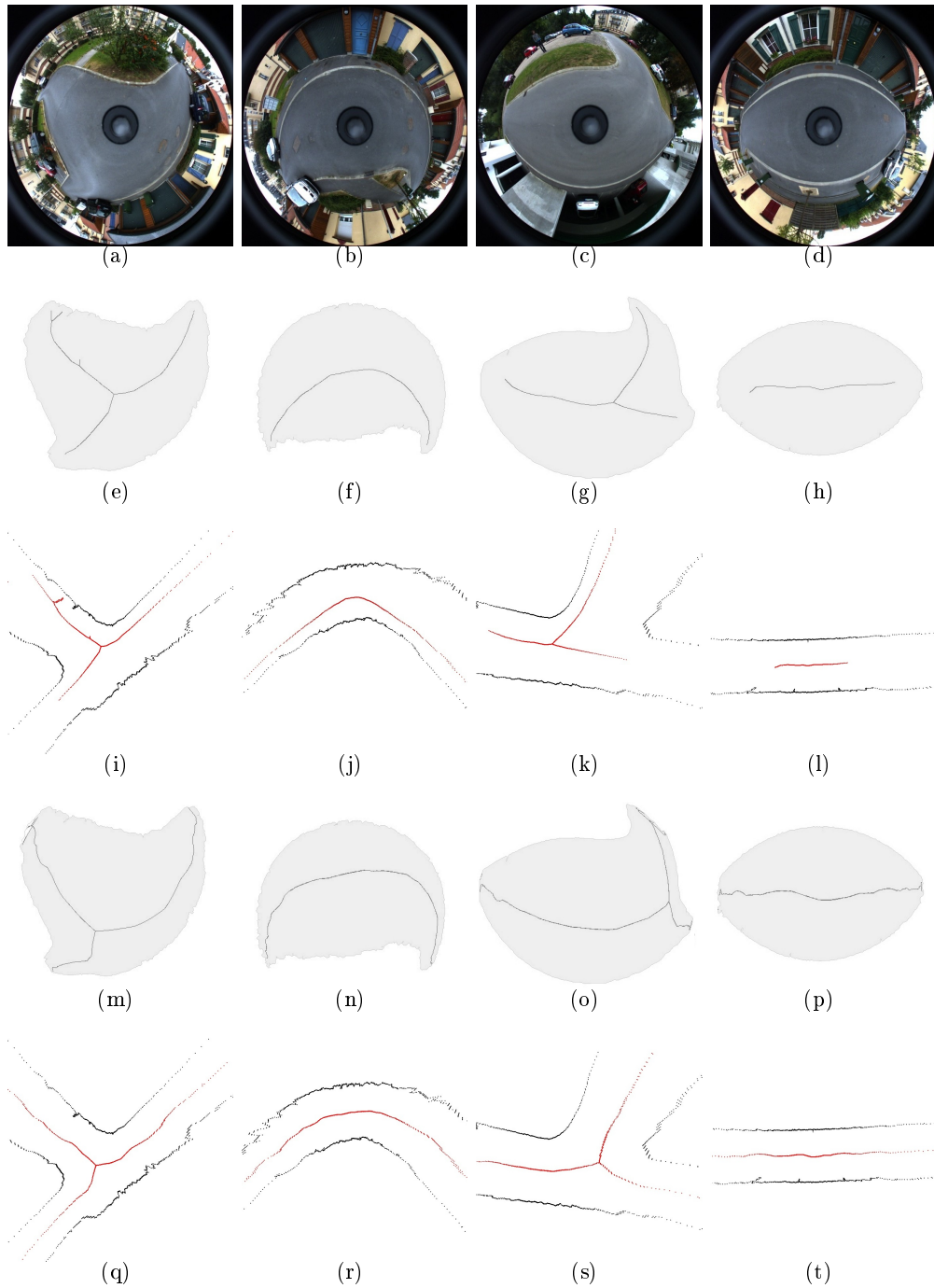


FIGURE 3.21 – Résultats comparatifs entre les versions standard et omnidirectionnelles du DMA. (a)-(d) : image capturée par le robot. (e)-(h) : squelette de l'espace libre calculé avec la version standard du DMA. (i)-(l) : reprojection dans le plan du sol des points du *DMA* calculé. (m)-(p) : squelette de l'espace libre calculé avec la version omnidirectionnelle du DMA. (q)-(t) : reprojection dans le plan du sol des points du *ODMA*.

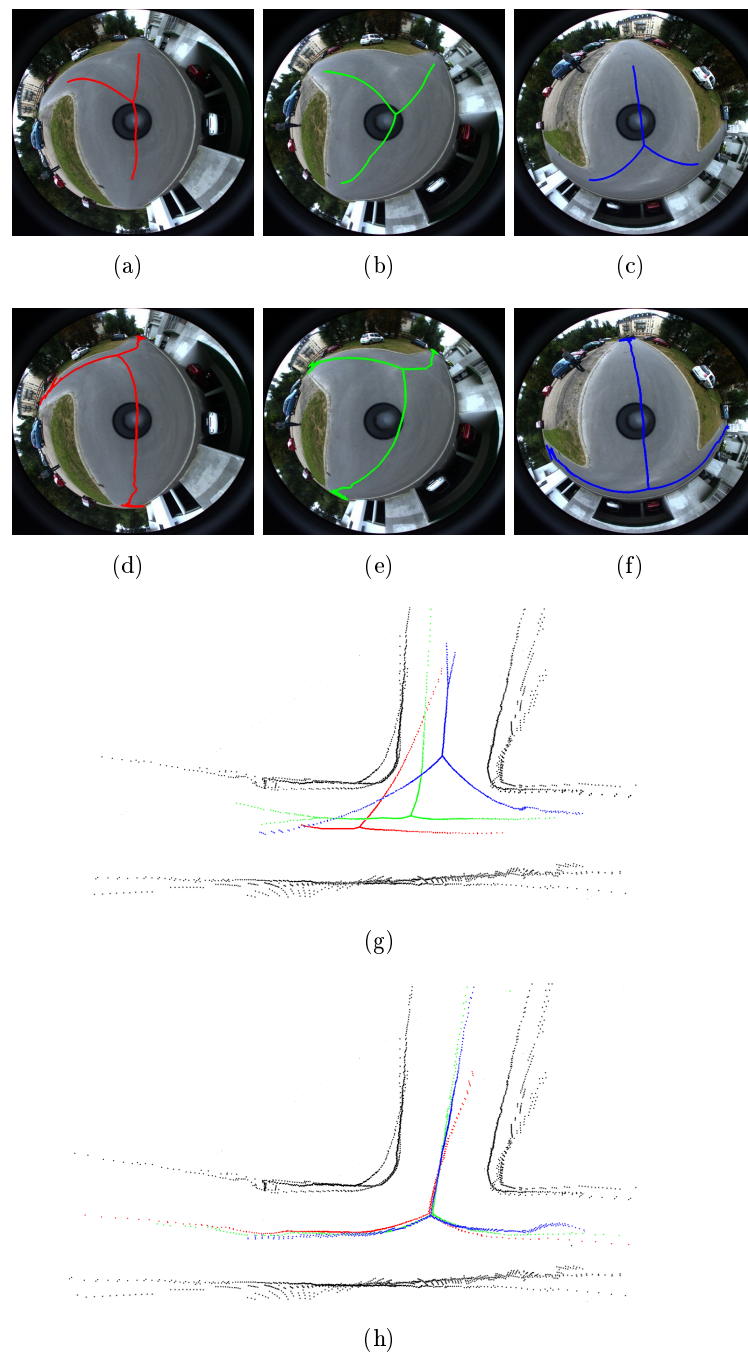


FIGURE 3.22 – Squelettes obtenues pour trois vues différentes d'un même carrefour, avec (a)(b)(c) la version standard du *Delta Medial Axis*, et (d)(e)(f) sa version adaptée. (g) et (h) : Reprojection dans le plan du sol des squelettes obtenus pour chacune des deux versions. A noter que pour la version standard, la valeur de  $\delta$  a été optimisée alors que pour la version adaptée, son réglage est automatique.





# Signature visuelle par invariant intégral

---

## Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>59</b>
<b>4.2</b>	<b>Indexation de lieu par construction de signature</b>	<b>60</b>
4.2.1	La notion de lieu	60
4.2.2	Invariance, discriminance et complétude	62
4.2.3	Etat de l'art	62
<b>4.3</b>	<b>Invariants intégraux et signature visuelle</b>	<b>65</b>
4.3.1	Principe général et formalisme de Haar	65
4.3.2	Utilisations existantes	65
<b>4.4</b>	<b>Elaboration de la signature</b>	<b>66</b>
4.4.1	Introduction	66
4.4.2	La configuration $\mathbf{x}$	67
4.4.3	Le groupe de transformation $\mathcal{G}$	68
4.4.4	Le noyau $f$	71
4.4.5	Expression finale de la signature	78
<b>4.5</b>	<b>Calcul effectif, implémentation et optimisations</b>	<b>79</b>
4.5.1	Analyse préalable : les différentes étapes	79
4.5.2	Calculs réalisés hors ligne	79
4.5.3	Optimisation des calculs en ligne	80
<b>4.6</b>	<b>Mesure de distance entre signatures</b>	<b>81</b>
<b>4.7</b>	<b>Validation expérimentale</b>	<b>83</b>
4.7.1	Avant-propos	83
4.7.2	Scénario en intérieur	83
4.7.3	Scénario en extérieur	89
<b>4.8</b>	<b>Conclusion</b>	<b>93</b>

---

## 4.1 Introduction

La représentation construite dans le chapitre précédent est parfaitement adaptée pour guider le robot le long des voies navigables existantes. Elle n'est en revanche pas suffisante pour lui permettre de se localiser dans son environnement. Il s'agit



FIGURE 4.1 – Définition de la portion de l'image utilisée dans le processus de construction de signature visuelle. (a) Image omnidirectionnelle acquise. (b) Portion de l'image qui sert au calcul des signatures.

pourtant d'une tâche fondamentale, car pour mener à terme une mission d'exploration, le robot doit pouvoir détecter lorsqu'il rejoint un lieu préalablement visité (fermeture de boucle).

Ce chapitre s'intéresse à cette problématique et propose une seconde représentation, basée apparence photométrique, qui sert de descripteur de lieu dans un processus de localisation. Fondée sur la théorie de l'invariance (invariants intégraux de Haar), et tenant compte des particularités géométriques du capteur utilisé, elle exploite le complément de l'espace libre afin de sélectionner dans l'image les données photométriques les plus discriminantes. (voir Fig. 4.1).

Ce chapitre commencera par aborder la question de l'indexation de lieu par construction d'invariant, en exposant un tour d'horizon des solutions existantes dans la littérature. Dans un second temps, la signature proposée sera détaillée, chacun des choix menant à sa formation étant présenté, discuté et justifié.

## 4.2 Indexation de lieu par construction de signature

### 4.2.1 La notion de lieu

D'après la définition du dictionnaire Larousse, le terme *lieu* désigne *une situation spatiale de quelque chose, ou quelqu'un, permettant de le localiser, de déterminer une direction, une trajectoire*. Autrement dit, deux positions géographiques différentes appartiennent au même lieu si l'information perceptive qui les définit aboutit à une localisation commune.

A partir de ce point, nous nommerons *configuration de  $p$* , notée  $\mathbf{c}_p$ , l'ensemble des informations perçues par les différents capteurs du robot pour une pose  $\mathbf{p}$  donnée. Nous notons  $E$  l'ensemble des dites configurations, et considérons un groupe de transformation  $\mathcal{G}$  agissant sur cet ensemble (un élément  $g$  de  $\mathcal{G}$  est donc une application de  $E$  dans  $E$ ). Deux configurations  $\mathbf{c}_p$  et  $\mathbf{c}_q$  de  $E$  partagent une relation

d'équivalence relativement à  $\mathcal{G}$  si et seulement si :

$$\exists g \in \mathcal{G} \mid \mathbf{c}_p = g \cdot \mathbf{c}_q \quad (4.1)$$

Pour chaque configuration  $\mathbf{c}_p$ , la relation 4.1 permet de définir une classe d'équivalence, dénommée *orbite*, et notée  $O(\mathbf{c}_p)$  :

$$O(\mathbf{c}_p) = \{c_q \in E \mid \exists g \in \mathcal{G} \mathbf{c}_q = g \cdot \mathbf{c}_p\} \quad (4.2)$$

Le groupe de transformation  $\mathcal{G}$  étant supposé vérifier une action de groupe, il est muni d'une loi de composition interne (notée  $\circ$ ) qui possède les propriétés suivantes :

- associativité :

$$\forall g_1, g_2, g_3 \in \mathcal{G}, g_1 \circ (g_2 \circ g_3) = (g_1 \circ g_2) \circ g_3$$

- existence d'un élément neutre :

$$\exists e \in \mathcal{G} \mid \forall g \in \mathcal{G}, e \circ g = g \circ e = g$$

- existence d'un inverse pour chaque élément :

$$\forall g \in \mathcal{G}, \exists g' \in \mathcal{G} \mid g \circ g' = g' \circ g = e$$

De ces propriétés découlent deux caractéristiques fondamentales relatives aux orbites :

$$\forall \mathbf{c}_p, \mathbf{c}_q \in E, \begin{cases} \mathbf{c}_q \in O(\mathbf{c}_p) \longrightarrow O(\mathbf{c}_p) = O(\mathbf{c}_q) \\ \mathbf{c}_q \notin O(\mathbf{c}_p) \longrightarrow O(\mathbf{c}_p) \cap O(\mathbf{c}_q) = \emptyset \end{cases} \quad (4.3)$$

Autrement dit, dans le cas où deux configurations sont liées par une relation d'équivalence, leurs orbites respectives sont confondues, alors que dans le cas contraire, elles sont disjointes. L'ensemble des orbites forme donc un partitionnement de l'espace des configurations. Soient  $\mathbf{p}$  et  $\mathbf{q}$  deux poses du robot, telles que  $\mathbf{p}$  appartienne à un lieu  $\mathcal{L}$ . On a :

$$\mathbf{q} \in \mathcal{L} \iff \mathbf{c}_q \in O(\mathbf{c}_p) \quad (4.4)$$

Le processus d'indexation de lieu vise donc à déterminer pour chaque pose de l'environnement, l'orbite dans l'espace des configurations à laquelle elle peut être associée. Dans la pratique, cette tâche est la plupart du temps impossible à réaliser. L'idée est plutôt, pour chaque configuration, de calculer une quantité, ou un ensemble de quantités, nommée signature, qui soit 1) identique pour deux configurations d'une même orbite, et 2) différente dans le cas contraire. L'indexation consiste alors à comparer la signature construite avec celles des configurations connues, ce qui permet de déterminer à quelle orbite elle appartient, et donc, par extension, le lieu dans lequel se trouve le robot. Tout l'enjeu est donc de créer une signature qui vérifie les deux propriétés citées juste avant, à savoir l'invariance au sein d'une orbite, et la discriminance permettant de distinguer deux d'entre elles.

### 4.2.2 Invariance, discriminance et complétude

L'invariance, sous-entend que la signature  $\mathcal{S}_p$  construite à partir de la configuration  $\mathbf{c}_p$  est identique à l'ensemble des signatures  $\mathcal{S}_q$  obtenues à partir des différentes configurations  $\mathbf{c}_q$  de la même orbite. Autrement dit :

$$\mathbf{c}_q \in O(\mathbf{c}_p) \longrightarrow \mathcal{S}_p = \mathcal{S}_q \quad (4.5)$$

De cette façon, la signature  $\mathcal{S}_{\mathcal{L}}$  d'un lieu  $\mathcal{L}$  peut être obtenue à partir d'une seule pose  $\mathbf{p}$ , tout en permettant une localisation valable pour l'ensemble du lieu :

$$\mathcal{S}_{\mathcal{L}} = \mathcal{S}_p \longrightarrow \forall \mathbf{q} \in \mathcal{L}, \mathcal{S}_q = \mathcal{S}_{\mathcal{L}} \quad (4.6)$$

Pour garantir sa qualité, il faut également s'assurer que deux lieux différents produisent deux signatures distinctes : on parle alors de discriminance. Pour deux poses  $\mathbf{p}$  et  $\mathbf{q}$  appartenant respectivement aux lieux  $\mathcal{L}_p$  et  $\mathcal{L}_q$ , on a :

$$\mathcal{S}_p = \mathcal{S}_q \longrightarrow \mathcal{L}_p = \mathcal{L}_q \quad (4.7)$$

Une signature à la fois invariante et discriminante est dite complète. La relation suivante est alors vérifiée :

$$\mathcal{L}_p = \mathcal{L}_q \iff \mathcal{S}_p = \mathcal{S}_q \quad (4.8)$$

### 4.2.3 Etat de l'art

Dans la pratique, il est toutefois très difficile de produire une signature qui vérifie ces deux propriétés. La plupart des travaux s'attachent donc à proposer des solutions qui cherchent le meilleur compromis entre invariance et discriminance, afin de maximiser les performances de l'indexation. Nous proposons maintenant d'en faire un tour d'horizon, en nous limitant toutefois aux méthodes qui sont fondées sur l'information visuelle, le capteur catadioptrique étant, dans ce travail, la seule modalité de perception utilisée.

Elles sont traditionnellement réparties en deux grandes familles :

- les approches locales, qui fondent la signature construite uniquement sur une portion de l'image sélectionnée à l'aide d'un ou plusieurs critères de pertinence ;
- les approches globales, qui représentent la scène perçue par une signature prenant en compte la totalité de l'image acquise.

#### 4.2.3.1 Les approches locales

Pour ce qui est des approches locales, elles procèdent généralement en deux étapes. Il s'agit dans un premier temps d'extraire dans l'image un ensemble de primitives géométriques (points, régions, ...) à partir de critères variés. Chacune d'elle est ensuite caractérisée par une empreinte (une signature), qui permet de la

reconnaître dans n'importe quelle prise de vue où elle apparaît. La signature de lieu est donc formée de l'ensemble des empreintes susceptibles d'être détectées lorsqu'une image  $y$  est acquise.

Une solution très populaire [Booij 2007, Se 2002, Andreasson 2005] passe par l'utilisation des descripteurs SIFT (*Scale Invariant Feature Transform*) [Lowe 1999, Lowe 2004]. Cet outil propose en effet la combinaison d'un processus de détection performant (fondé sur la recherche d'extrema locaux dans une différence de gaussiennes multi-échelle appliquée à l'image) associé à un descripteur à fort pouvoir discriminant qui en outre est invariant à de nombreuses transformations (rotation, translation, illumination, changement d'échelle, ...). Dans [Booij 2007], la similarité entre deux images (panoramiques) est déterminée par le ratio entre le nombre de points SIFT mis en correspondance qui vérifient la contrainte épipolaire, et le nombre de points SIFT détectés. Plus ce ratio est élevé, plus les deux images sont susceptibles de provenir du même lieu. Dans [Se 2002], les auteurs tirent profit de l'échelle et l'orientation qui sont associées à chaque point SIFT extrait afin de déterminer la structure de l'environnement observé par stéréovision. Dans [Andreasson 2005], les descripteurs SIFT sont associés à une localisation *Monte-Carlo* pour développer une solution adaptée aux images omnidirectionnelles qui tient compte de ses spécificités.

Bien entendu, de nombreuses alternatives existent. L'utilisation de points d'intérêt peut ainsi passer par d'autres descripteurs (SURF [Valgren 2010], MSIFT [Tamimi 2006, Valgren 2006], ..). Certains travaux considèrent également des primitives géométriques qui ne sont pas des points. Dans [Scaramuzza 2008], l'approche consiste à extraire les droites radiales visibles dans une image omnidirectionnelle (transformée de Hough), puis à les décrire à partir d'histogrammes de gradients calculés sur leurs voisinages. Dans [Tuytelaars 2004, Matas 2002, Ramisa 2009], ce sont des régions entières qui sont isolées puis décrites. Enfin, dans [Goedeme 2004], moyennant un positionnement précis du système d'acquisition (caméra perspective traditionnelle dont le plan image est perpendiculaire au sol), et supposant des éléments verticaux dans la scène observée, l'approche consiste à extraire les segments reliant chaque couple de points d'intérêt appartenant à la même colonne (ce sont donc des segments verticaux dans la scène), puis à les caractériser à partir de leur voisinage.

En terme de performance, ces approches présentent généralement des capacités d'invariance et de discriminance élevées, mais au prix de temps de calcul et d'occupation mémoire importants. En outre, la plupart d'entre elles sont robustes à de nombreux éléments, notamment à des changements d'illumination (d'amplitude modérée), d'échelle, ainsi qu'à d'éventuelles occultations. Elles sont néanmoins complexes à utiliser dans un processus d'indexation de lieu, car fortement dépendantes de la phase de détection. Les primitives détectées dans deux images issues d'un même lieu ne sont pas forcément toujours les mêmes, et nécessitent donc un processus de mise en correspondance adapté.

### 4.2.3.2 Les approches globales

De leur côté, les approches globales construisent une signature qui tient compte de l'image dans son ensemble. De cette façon, sa dimensionalité est constante quelle que soit l'image acquise, ce qui rend le processus de comparaison entre signatures plus simple. Pour la même raison, ces approches sont généralement beaucoup plus rapides et clairement moins coûteuses en mémoire. En contrepartie, elles ont un pouvoir discriminant moins élevé, et sont sensibles aux éventuelles occultations.

Une première solution populaire entrant dans cette catégorie consiste à caractériser l'image acquise à travers un ensemble de valeurs formant un (ou plusieurs) histogramme(s). De cette façon, le processus de mise en correspondance peut passer par l'utilisation d'une des nombreuses métriques existantes pour estimer la distance entre distributions. Dans [Ulrich 2000], un ensemble de 6 histogrammes est utilisé. Chacun d'entre eux décrit, de manière indépendante, la distribution d'un des canaux porté par les espaces de couleur RGB et HSV. Dans [Kosecka 2004, Gonzalez-Barbosa 2002], les gradients sont mis à contribution. Dans [Kosecka 2004], la signature est un histogramme contenant la répartition des orientations des gradients sur l'ensemble de l'image pondérés par leur amplitude. Dans [Gonzalez-Barbosa 2002], un choix empirique de 5 caractéristiques locales obtenues par dérivations successives suivant différentes orientations permet de construire un ensemble d'histogrammes indépendants servant là encore de descripteur. Enfin, dans [Zhou 2003], de nombreuses caractéristiques différentes (texture, couleur, contours, ...) sont évaluées en chaque pixel et utilisées de concert pour former un histogramme multidimensionnel qui sert de signature.

D'autres travaux proposent une approche fondée sur l'analyse fréquentielle de l'image. Dans [Ishiguro 1996, Menegatti 2004], l'image omnidirectionnelle utilisée est portée sur une représentation cylindrique. Pour chaque ligne de cette dernière, la transformée de Fourier est calculée et les 15 premiers coefficients sont retenus. La signature est alors la matrice des coefficients obtenus pour chaque ligne de la représentation. Pajdla [Pajdla 1999] quant à lui, propose l'estimation d'une orientation de référence d'images omnidirectionnelles à partir de ce qu'il nomme *Zero Phase Representation* (Représentation de phase nulle). Sa méthode permet ainsi d'aligner les dites images en utilisant la phase du premier coefficient de sa transformée de Fourier.

Une autre approche qui a donné lieu à une riche littérature est l'utilisation des descripteurs *gist* (littéralement "l'essentiel"). La première version de l'outil est proposée par Oliva et Torralba [Oliva 2001], qui comparent deux descripteurs fondés sur la transformée de Fourier d'une image en niveau de gris. Dans le premier cas, la transformée est calculée sur l'ensemble de l'image tandis que dans le second, l'image est préalablement découpée en une grille grossière (8x8), chacune de ces cases étant traitée indépendamment. Ils montrent alors qu'une telle subdivision permet d'extraire une quantité d'information supérieure, et l'utilisent pour définir différentes propriétés dans l'image. Depuis, l'approche a été reprise dans de nombreux travaux qui se différencient principalement par la taille de la grille utilisée et surtout la façon

dont est caractérisée chacune des régions. Ainsi, dans [Torralba 2003, Oliva 2006], un ensemble de filtres orientables (différentes échelles et différentes orientations) sont appliqués puis moyennés sur des grilles de tailles variables. Dans [Pugeault 2010], ce sont cette fois des ondelettes de Gabor qui sont choisies pour définir la scène perçue. Dans [Siagian 2007], en plus des filtres de Gabor, chaque région est décrite à partir de l'information couleur, et de l'intensité globale. A noter enfin que certains travaux [Murillo 2009, Singh 2010] ont appliqué avec succès cette approche dans le cadre d'images omnidirectionnelles (en découpant chaque image en 4 sous-régions considérées comme autant d'images indépendantes).

## 4.3 Invariants intégraux et signature visuelle

### 4.3.1 Principe général et formalisme de Haar

Dans la continuité des travaux de Charron [Charron 2006], nous élaborons une signature qui s'appuie sur une formalisation mathématique de la notion d'invariant de lieu. Intuitivement, chaque lieu étant caractérisé par une orbite dans l'espace des configurations, il suffit d'intégrer une fonction le long de cette dernière pour obtenir une quantité invariante. Un changement de pose au sein du lieu a en effet pour seule conséquence de décaler les bornes de l'intégration, l'orbite dans son ensemble étant quoi qu'il arrive considérée.

Pour construire une telle signature, nous utilisons le formalisme des invariants intégraux de Haar, qui s'écrit :

$$I_{Haar} = \frac{1}{|\mathcal{G}|} \int_{\mathcal{G}} f(g.\mathbf{x}) dg, \text{ avec } g \in \mathcal{G}, \text{ et } |\mathcal{G}| = \int_{\mathcal{G}} dg \quad (4.9)$$

En appliquant chaque élément  $g$  d'un groupe de transformation  $\mathcal{G}$ , à une configuration  $\mathbf{x}$ , l'ensemble des configurations  $g.\mathbf{x}$  appartenant à la même orbite est défini. Une fonction noyau  $f$  (définie positive) quelconque est alors intégrée sur l'ensemble de cette orbite. La valeur obtenue est finalement normalisée par  $|\mathcal{G}|$ , qui peut être vu comme le nombre de configurations formant l'orbite.

Cet invariant est un scalaire, ce qui rend compréhensible la critique généralement émise à l'encontre de cette approche : son faible pouvoir discriminant. Toutefois, la fonction noyau étant quelconque, rien n'empêche d'en utiliser une collection, ce qui permet de construire une famille d'invariants qui aboutit à une signature aux performances supérieures.

### 4.3.2 Utilisations existantes

Plusieurs travaux existants ont déjà utilisé cette stratégie dans un processus de construction de signature, que ce soit dans un contexte de localisation en robotique [Wolf 2002, Charron 2006, Labbani-Igbida 2011], ou plus généralement dans une application de reconnaissance d'images basée sur le contenu [Schulz-mirbach 1996, Siggelkow 2002, Halawani 2004].



Dans [Schulz-mirbach 1996], les auteurs cherchent à définir une stratégie efficace de reconnaissance d'images. Ils considèrent une invariance relative au groupe des rotations et translations dans le plan, et proposent une signature fondée sur une collection de 12 fonctions noyaux correspondant à des monômes d'ordre 1 à 3. Ce travail est ensuite repris par Siggelkow [Siggelkow 2002], qui, dans un contexte similaire, considère le même groupe de transformations en introduisant cette fois un partitionnement flou de l'espace des configurations. Chaque fonction noyau utilisée est convoluée à un ensemble de fonctions triangle formant une famille d'intégrales se comportant comme une distribution. La signature ainsi obtenue est un ensemble d'histogrammes (un par fonction noyau utilisée) qui a pour conséquence d'augmenter considérablement le pouvoir discriminant.

Cette même approche est appliquée dans un contexte de robotique mobile dans [Wolf 2002], où, couplée à une localisation Monte-Carlo, elle permet à un robot de se situer dans son environnement. Plus récemment, Charron [Charron 2006, Labbani-Igbida 2011] propose d'utiliser le formalisme de Haar sur des images omni-directionnelles, également dans un contexte de localisation en robotique mobile. Le groupe de transformations est alors adapté en conséquence, et, tout en conservant l'idée des histogrammes flous suggérés par Siggelkow, le noyau utilisé (fondé sur une différence de Gaussiennes), est cette fois adapté à la géométrie particulière du capteur.

## 4.4 Elaboration de la signature

### 4.4.1 Introduction

Bien que la fonction noyau proposée par Charron [Charron 2006] offre des performances de qualité en environnement intérieur, son pouvoir discriminant n'est pas suffisant en extérieur pour éviter des situations où de fausses fermetures de boucle sont détectées. Nous proposons donc une nouvelle signature, qui tire pleinement partie de l'information à disposition : non seulement chaque canal  $R, G, B$  est considéré, mais en plus l'espace libre extrait est mis à contribution dans sa construction. La formulation de l'équation (4.9) met en évidence la nécessité de définir trois éléments principaux, qui sont :

- la configuration  $\mathbf{x}$  : construite à partir de l'image, elle doit tenir compte de la géométrie particulière du capteur, ainsi que des particularités liées au système dans son ensemble ;
- le groupe de transformation  $\mathcal{G}$  : directement lié à la manipulation de la configuration construite, il doit effectuer le lien entre les déplacements réels du robot et les modifications résultantes dans l'espace des configurations ;
- la (ou les) fonction(s) noyau  $f$  : clé de voûte de la signature résultante, elle doit capturer les propriétés spécifiques du lieu observé afin de rendre la signature construite la plus discriminante possible.

Nous proposons maintenant de détailler et de justifier les choix retenus pour chacun d'entre eux. Notons que ces différents éléments sont interdépendants, mais que, pour la structuration et la clarté du discours, ils seront étudiés séparément.

#### 4.4.2 La configuration $\mathbf{x}$

##### 4.4.2.1 D'une configuration à toutes les autres

A un instant  $t$ , l'image  $I_t$  acquise par le capteur omnidirectionnel regroupe l'ensemble des informations à disposition du robot concernant le lieu dans lequel il évolue. Il est donc clair que la configuration  $\mathbf{x}$  dérive directement de  $I_t$ . En outre, le processus d'intégration nécessite, pour une orbite donnée, d'accéder à l'ensemble des configurations  $\mathbf{y} = g.\mathbf{x}$ . Dans un cas idéal, il faudrait donc posséder, pour chaque lieu connu, la collection des images pouvant être acquises à chacune des poses le constituant. C'est bien entendu impossible (aussi bien pour des considérations de mémoire, que pour des aspects pratiques). L'idée est donc de construire une configuration  $\mathbf{x}$  à partir de laquelle chaque configuration  $g.\mathbf{x}$  puisse être déduite. Nous proposons ici une représentation qui passe par la projection de  $I_t$  sur la sphère d'équivalence.

##### 4.4.2.2 Sphère d'équivalence et paramétrisation $(\theta, \varphi)$

Dans le chapitre 2, nous avons vu qu'il était possible de modéliser la projection d'un point 3D dans l'image par une double projection, d'abord sur la sphère unitaire, puis sur le plan image normalisé [Barreto 2001]. Nous nous intéressons en particulier ici à la seconde, notée  $f_{img}$ , qui n'est rien d'autre que la projection de la surface d'une sphère sur un plan. Pour rappel, les équations de projection sont les suivantes :

$$\mathbf{p} = f_{img}(\mathbf{P}_S), \text{ avec } \begin{cases} \mathbf{x} = \frac{X_S}{Z_S + \xi} \\ \mathbf{y} = \frac{Y_S}{Z_S + \xi} \end{cases} \quad (4.10)$$

Le point  $P_S$  peut être exprimé en coordonnées sphériques de type rayon-colatitude-longitude, avec  $\theta$  la colatitude et  $\varphi$  la longitude (voir Fig.4.2(a)). La même projection de la sphère vers le plan image normalisé s'exprime alors par :

$$\mathbf{p} = f_{img}(\mathbf{P}_S), \text{ avec } \begin{cases} \mathbf{x} = \frac{\sin \theta \cos \varphi}{\cos \theta + \xi} \\ \mathbf{y} = \frac{\sin \theta \sin \varphi}{\cos \theta + \xi} \end{cases} \quad (4.11)$$

En ajoutant le passage du plan image normalisé vers les coordonnées pixelliques, on obtient :

$$\mathbf{p} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ 1 \end{pmatrix} = \frac{\sin \theta}{\cos \theta + \xi} \cdot \mathbf{K} \cdot \begin{pmatrix} \cos \varphi \\ \sin \varphi \\ 1 \end{pmatrix} \quad (4.12)$$

où  $\mathbf{K}$  désigne toujours la matrice des paramètres intrinsèques. A partir de ces équations, il est donc possible de réaliser un échantillonnage régulier de la sphère

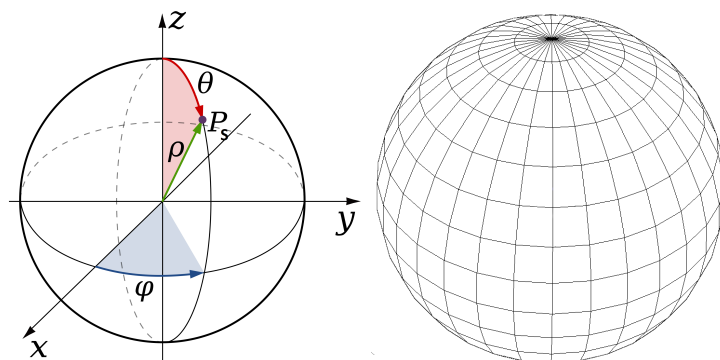


FIGURE 4.2 – (a) Convention utilisée  $(\rho, \theta, \varphi)$  pour le passage en coordonnées sphériques. (b) Exemple d'échantillonnage de la sphère.

(voir Figure 4.2(b)) et d'y apposer l'information photométrique correspondante de l'image. Les coordonnées  $(\mathbf{u}, \mathbf{v})$  obtenues pour un couple  $(\theta, \varphi)$  donné ne sont en général pas entières. Une interpolation bilinéaire est donc utilisée pour déterminer les composantes  $R, G, B$  les plus adaptées. En outre, pour une portion non négligeable de la sphère (la quasi-totalité de la demi-sphère supérieure), les coordonnées  $(\mathbf{u}, \mathbf{v})$  calculées dépassent le cadre de l'image. Dans ce cas, le couple  $(\theta, \varphi)$  appartient à la zone inutile et est marqué comme tel afin de ne pas être pris en compte dans le processus d'intégration. Enfin, pour des facilités de représentation et de manipulation, la configuration  $\mathbf{x}$  est dans la pratique définie sur le plan  $\Pi(\theta, \varphi)$  tel que présenté Figure.4.3.

### 4.4.3 Le groupe de transformation $\mathcal{G}$

#### 4.4.3.1 Première réflexion

Choisi en adéquation avec la configuration construite, le groupe de transformation définit l'invariance que l'on souhaite donner à la signature, c'est-à-dire les événements (susceptibles de modifier la perception) à prendre en compte dans le processus d'intégration. Ils peuvent être de deux natures :

- 1) une modification de la scène en elle-même, que ce soit des objets qui ont changé de position ou simplement disparus, ou encore la scène dans son ensemble qui est éclairée différemment,
- 2) une variation dans la perception qu'en a le robot, provoquée par un changement de point de vue lorsqu'il se déplace.

Similairement à [Charron 2006], nous proposons de choisir un groupe de transformation qui ne considère que la seconde catégorie. A noter toutefois, que, même sans les prendre en compte directement, la signature que nous construisons permet de considérer une partie des changements qui surviennent dans un lieu. S'ils restent mineurs, ils n'ont en effet que peu d'impact sur la signature finale, puisque leur contribution dans le processus d'intégration reste négligeable.



(a)



(b)

FIGURE 4.3 – De l'image brute (a) au plan  $\Pi(\theta, \varphi)$  (b). Seule la portion de la sphère correspondant à l'image projetée est représentée.

Soient  $t_x, t_y$  les translations et  $\phi$  les rotations du robot dans le plan du sol. L'intégrale de Haar s'écrit alors :

$$I_{haar}(\mathbf{x}) = \frac{\int_{t_x, t_y, \phi} f(g\mathbf{x}) dt_x dt_y d\phi}{\left| \int_{t_x, t_y, \phi} dt_x dt_y d\phi \right|} \quad (4.13)$$

Comme expliqué précédemment, il est inenvisageable de déterminer l'ensemble des configurations  $\mathbf{y} = g\mathbf{x}$ . L'idée est donc d'intégrer non plus sur l'ensemble des déplacements du robot, mais plutôt sur l'ensemble des transformations que peut subir la configuration.

#### 4.4.3.2 Représentations sensorimotrices [Charron 2006]

Dans sa thèse, Charron [Charron 2006] considère et analyse trois approches différentes.

La première étudie l'effet des déplacements du robot sur les points 3D  $(X, Y, Z)$ , et en déduit par projection les modifications induites dans l'image. L'intégrale de Haar peut alors y être directement exprimée en appliquant un changement de variable. On trouve :

$$\mathcal{I}_{haar}(\mathbf{x}) = \frac{\int_{u,v} f(g\mathbf{x})|J|dudv}{\int_{u,v} dudv} \quad (4.14)$$

où  $|J|$  est le déterminant du jacobien lié au changement de variable qui décrit la relation entre l'élément de surface dans la scène et celui dans l'image. Les dérivées partielles sont calculées à partir du modèle de projection paraboloidale. Ce dernier n'étant pas celui de la sphère d'équivalence, le jacobien obtenu n'a pas d'intérêt dans le contexte présent. En revanche, il n'est pas défini partout dans l'image, et présente une singularité au niveau de la ligne d'horizon, rendant l'impact des points au centre de l'image négligeable par rapport à ceux en périphérie dans le processus d'intégration. La signature ainsi obtenue n'est pas stable, ce qui la rend dans la pratique difficilement utilisable.

La deuxième solution consiste à projeter le monde sur la sphère unitaire et à considérer non plus les déplacements du robot dans la scène, mais les déplacements des points sur cette sphère. Cette approche implique que lorsque le robot change de pose, l'information présente sur la sphère reste identique. Autrement dit, les points s'y déplacent mais n'en sortent pas. Pour une rotation du robot, c'est parfaitement exact, puisque les points projetés subissent une simple rotation autour de l'axe vertical (donc un décalage suivant  $\varphi$ ). Pour une translation, c'est en revanche une approximation. Ces mêmes points subissent en effet une transformation sur la sphère composée d'un déplacement suivant  $\theta$  ( $dg_\theta = d\theta$ ) et d'un déplacement suivant  $\varphi$  fonction de  $\theta$  ( $dg_\varphi = \sin(\theta)d\varphi$ ). Nous considérons néanmoins que ces transformations vérifient toujours une action de groupe (faute de quoi l'invariance n'est plus valable). Le calcul est ensuite porté dans le plan image par l'intermédiaire d'un nouveau jacobien, qui cette fois ne contient pas de pôle. Cette solution présente néanmoins le défaut de nécessiter la reprojexion des points de la sphère vers le plan image.

Pour cette raison, la solution finale retenue par Charron, et que nous avons également sélectionné, est de travailler directement sur la sphère d'équivalence. L'invariant de Haar correspondant est alors fondé sur l'intégration suivant les coordonnées sphériques (avec  $\rho = cste$ ) :

$$\mathcal{I}_{haar}(\mathbf{x}) = \frac{\int_{\theta,\varphi} f(g\mathbf{x}) \sin \theta d\theta d\varphi}{\int_{\theta,\varphi} \sin \theta d\theta d\varphi} \quad (4.15)$$

#### 4.4.3.3 Construction des configurations $\mathbf{y} = g\mathbf{x}$

Soit  $\mathbf{y}$  une configuration obtenue par application d'une transformation  $g \in \mathcal{G}$  sur  $\mathbf{x}$ . En représentant la sphère d'équivalence par le plan  $(\theta, \varphi)$  (qui est supposé cyclique), la configuration  $\mathbf{y}$  peut être vue comme une version translatée de  $\mathbf{x}$ . Si  $g = (\Delta\theta, \Delta\varphi)$ , alors :

$$\begin{aligned} &\forall \theta \in [0, \pi], \varphi \in [0, 2\pi], \\ &\mathbf{y}(\theta', \varphi') = g.\mathbf{x}(\theta, \varphi) \text{ avec } \begin{cases} \theta' \equiv \theta + \Delta\theta \pmod{\pi} \\ \varphi' \equiv \varphi + \Delta\varphi \pmod{2\pi} \end{cases} \end{aligned} \quad (4.16)$$

Ainsi, quelle que soit la transformation  $g$  considérée, pour tout point  $\mathbf{y}(\theta', \varphi')$  il existe un point  $\mathbf{x}(\theta, \varphi)$  équivalent. En particulier :

$$\exists g \in G \mid \mathbf{x}(\theta, \varphi) = (g.\mathbf{x})(0, 0) = \mathbf{y}(0, 0) \quad (4.17)$$

Ainsi, en choisissant un noyau  $f$  à support local, par exemple  $f = \mathbf{x}(0, 0)$ , on peut écrire :

$$\int_{\theta, \varphi} g\mathbf{x}(0, 0)d\theta d\varphi = \int_{\theta, \varphi} \mathbf{x}(\theta, \varphi)d\theta d\varphi \quad (4.18)$$

Autrement dit, ce n'est plus la configuration  $\mathbf{x}$  qui subit  $g$ , mais plutôt le support de la fonction noyau. De cette façon, le processus d'intégration sur  $\mathcal{G}$  ne nécessite pas le calcul des différentes configurations  $\mathbf{y} = g\mathbf{x}$ , mais se limite à un parcours de  $\mathbf{x}$ . Bien sûr le noyau présenté pour l'exemple est insuffisant (l'intégrale de Haar ainsi formée correspond simplement à la moyenne photométrique de  $\mathbf{x}$ ). Néanmoins, dans la mesure où  $f$  est à support local, nous supposons l'approche valide.

#### 4.4.4 Le noyau $f$

##### 4.4.4.1 Première analyse

Bien que théoriquement n'importe quelle fonction intégrable à support compact puisse servir de noyau  $f$ , son choix est dans la pratique une étape cruciale de l'élaboration de la signature. D'un côté, elle doit être capable de capturer les propriétés intrinsèques au lieu perçu, afin de limiter au maximum les problèmes d'ambiguïté perceptive. De l'autre, elle doit, être à support compact.

Pour rappel, Charron utilise dans son travail un noyau fondé sur une différence de gaussiennes qui capture l'information de structure locale (les contours), mais perd l'ensemble du contenu photométrique. Bien qu'en intérieur ce choix ait donné des résultats convaincants, en extérieur les scènes manquent souvent de contours structurés, ce qui affaiblit fortement les performances de la méthode. Nous proposons ici un noyau original qui combine plusieurs composantes complémentaires :

- un noyau de pertinence qui permet de ne prendre en compte que l'information pertinente sur le domaine d'intégration ;
- un noyau structurel qui permet de pondérer l'information portée par l'image en fonction de la distance des objets projetés relativement au robot ;
- un noyau monomial qui code l'information photométrique locale des points de l'image ;
- un noyau de partition qui permet de découper l'espace photométrique afin de calculer une famille d'intégrales au lieu d'une seule.

##### 4.4.4.2 Le noyau de pertinence

Le processus d'extraction de l'espace libre présenté au début de ce mémoire permet de séparer dans l'image l'espace navigable  $FS$  et les obstacles  $\overline{FS}$ . Lors

de l'élaboration de la signature, nous nous sommes interrogés sur la pertinence de chacune de ces deux régions quant à la caractérisation de la scène perçue d'un point de vue photométrique.

D'un côté, le sol est une région globalement homogène qui a une probabilité importante d'être commune à plusieurs lieux (typiquement du goudron dans un scénario urbain). Pourtant, il occupe une portion non négligeable de l'image (donc de la configuration), ce qui le rend prépondérant dans une approche globale. Autrement dit, son impact dans la signature construite est majeur, alors qu'il est non seulement peu porteur d'information photométrique, mais qu'en plus cette information est partagée par plusieurs orbites.

De l'autre, les obstacles contiennent une information photométrique riche qui a en outre beaucoup moins de chance d'être commune à différents lieux.

Pour illustrer ces propos, la figure 4.4 propose pour trois prises de vue, une représentation séparée de  $FS$  et  $\overline{FS}$ . On constate aisément qu'en se basant uniquement sur la première ligne ( $FS$ ), il est impossible de déterminer si les images sont issues ou non d'un même lieu. A l'inverse, la seconde ligne seule permet de constater que la première et la dernière image sont prises à des positions proches, tandis que la seconde est visiblement issue d'un lieu totalement différent.

Dans le processus d'intégration, le but est par conséquent de ne considérer que les points de la configuration qui représentent un obstacle, sans pour autant modifier les bornes d'intégration. Pour cela, nous construisons le noyau de pertinence  $f_p(\theta, \varphi)$  sous la forme d'un coefficient égal à 1 si le couple  $(\theta, \varphi)$  est la projection d'un obstacle, et 0 sinon. Autrement dit :

$$f_p(\theta, \varphi) = \begin{cases} 1 & \text{si } (u, v) = f_{img}(\theta, \varphi) \in \overline{FS} \\ 0 & \text{sinon} \end{cases} \quad (4.19)$$

De cette façon, l'ensemble des couples  $(\theta, \varphi)$  est considéré, mais seuls ceux appartenant à  $\overline{FS}$  auront un impact sur la signature construite.

#### 4.4.4.3 Le noyau structurel

Le processus d'intégration modélisé par l'équation (4.13) prend en compte la résolution non uniforme de la grille qui échantillonne la sphère d'équivalence par l'intermédiaire du terme  $\sin(\theta)$ . En revanche, il ne considère à aucun moment la distance des objets perçus relativement au robot. Pourtant, les variations induites dans l'image lorsque le robot se déplace dépendent de la position des objets dans la scène.

Pour nous en convaincre, considérons le couple d'images de synthèse présenté Fig.4.5 (a) et (b). Le déplacement les séparant est une translation d'1m50. La surface occupée par chaque objet varie entre les deux images (c'est particulièrement visible pour la porte de couleur rouge), mais pas de façon identique. La conséquence directe est un changement important du contenu image (traduit par les histogrammes proposés Fig.4.5 (c) et (d)).

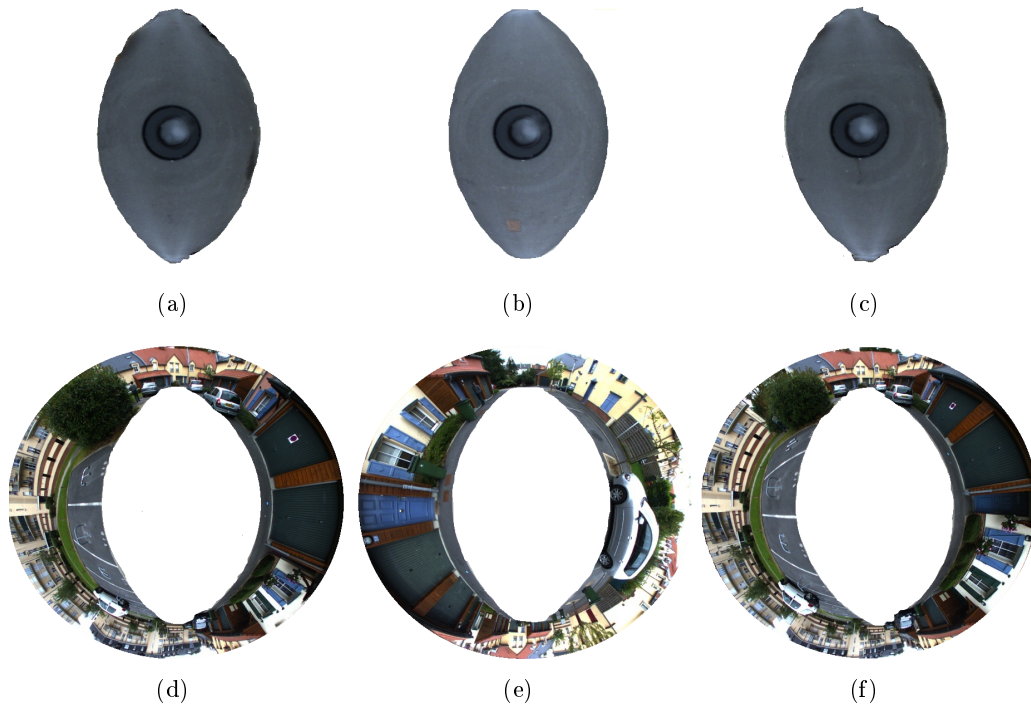


FIGURE 4.4 – Illustration de la pertinence de l'information photométrique portée par l'espace navigable et les obstacles dans un processus d'indexation de lieu. Première ligne (a,b,c) : le sol seul ne permet clairement pas d'affirmer une ressemblance entre deux prises de vue. Deuxième ligne (d,e,f) : les obstacles en revanche sont largement suffisants. On voit sans problème que les image (d) et (f) sont issues de positions voisines.

Partant de ce constat, nous avons cherché à compenser ces variations à travers un noyau structurel chargé d'intégrer la géométrie supposée de la scène dans le processus d'intégration.

Il est construit avec pour objectif de pondérer l'impact de chaque point en fonction de sa distance au robot. Nous faisons pour cela une hypothèse sur la scène perçue, et la supposons constituée d'un unique plan horizontal (le sol) intersectant un ensemble de plans verticaux au niveau des frontières de l'espace libre (les obstacles). Les points du sol partagent donc tous la même coordonnée  $Z$ . Cette hypothèse, tout à fait raisonnable en environnement intérieur, est plus discutable en environnement extérieur. Toutefois, comme nous le verrons par la suite, le noyau structurel proposé génère une amélioration visible sur l'un des scénarios en extérieur, ce qui nous donne un indice positif sur sa pertinence en général.

Nous mettons à profit l'unique information structurelle dont nous disposons sur la scène : les frontières de l'espace navigable. Le contour actif  $\Gamma$  les définissant est continu. Sa projection sur la sphère d'équivalence l'est donc également. Autrement dit :



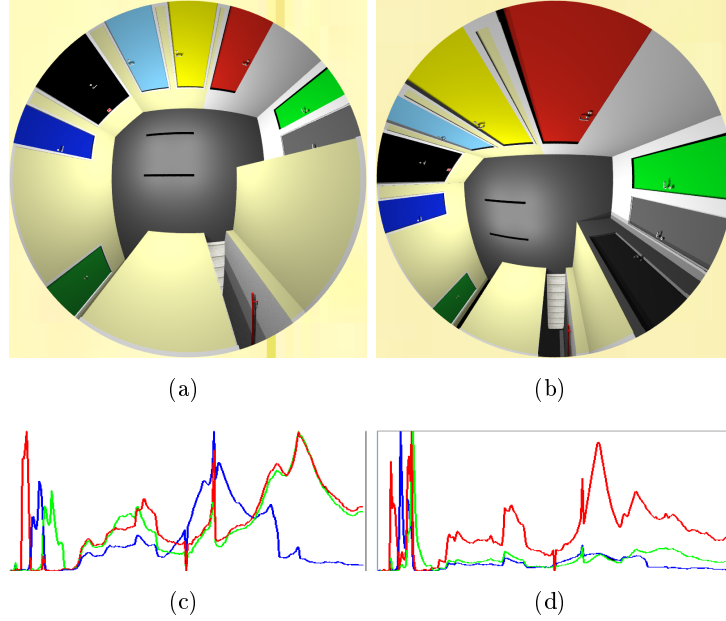


FIGURE 4.5 – (a) et (b) Couple d’images capturées par un capteur catadioptrique simulé dans un même lieu de synthèse (POV-Ray). (c) et (d) Histogrammes couleur de chacune des deux images.

$$\forall \varphi \in [0, 2\pi], \exists \theta \in [\pi/2, \pi] \mid f_{img}(\theta, \varphi) \in \Gamma \quad (4.20)$$

où  $f_{img}$  désigne la fonction qui projette un point de la sphère équivalente dans le plan image. Nous notons  $\mathbf{m}(\varphi)$  ce point (voir Figure.4.6). Comme vu dans le chapitre précédent,  $\Gamma$  est partie intégrante du sol.  $\forall \varphi \in [0, 2\pi]$ , la coordonnée  $Z = H$  est donc connue, ce qui permet de retrouver les coordonnées 3D du point  $\mathbf{M}(\varphi)$  correspondant :

$$\mathbf{M}(\varphi) = f_{sph}^{-1}(\mathbf{m}(\varphi)) \quad (4.21)$$

L’expression de  $f_{sph}^{-1}$  pour les points d’altitude  $Z = H$  est présentée dans le chapitre précédent, et s’exprime, pour un point de la sphère unité  $P_s = (\rho = 1, \theta, \varphi)$  par :

$$P = f_{sph}^{-1}(P_s) : \begin{cases} X = H \cos(\varphi) \tan(\theta) \\ Y = H \sin(\varphi) \tan(\theta) \\ Z = H \end{cases} \quad (4.22)$$

Le point  $\mathbf{M}(\varphi)$  pouvant être vu comme la projection au sol de l’ensemble des points 3D (non sol) appartenant à la même radiale sur la sphère, nous proposons de définir le noyau structurel pour chaque point  $(\theta, \varphi)$  comme la moyenne des distances entre  $\mathbf{M}(\varphi)$  et ses voisins directs ( $\mathbf{M}(\varphi + \Delta\varphi)$  et  $\mathbf{M}(\varphi - \Delta\varphi)$ ). Ces distances reflètent

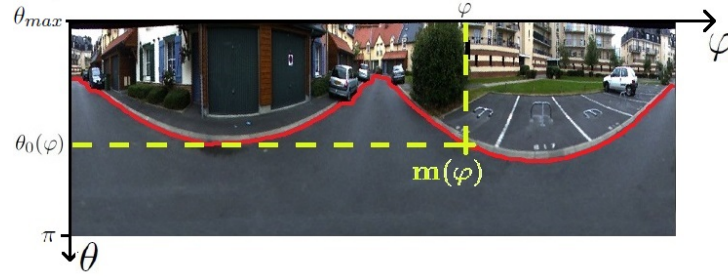


FIGURE 4.6 – Exemple (en jaune) de la projection sur le plan  $\Pi(\theta, \varphi)$  du point  $\mathbf{m}(\varphi)$  calculé pour un  $\varphi$  donné, qui correspond à la frontière de l'espace libre dans cette direction. Ce point caractérise donc la distance au robot de chaque point appartenant à la même radiale.

la structure de la scène et caractérisent les changements de proportions dans l'image qui sont liés à l'éloignement des objets observés (voir Figure 4.7). On a alors :

$$f_s(\mathbf{x}(\theta, \varphi)) = \frac{\|\mathbf{M}(\varphi) - \mathbf{M}(\varphi + d\varphi)\| + \|\mathbf{M}(\varphi) - \mathbf{M}(\varphi - d\varphi)\|}{2} \quad (4.23)$$

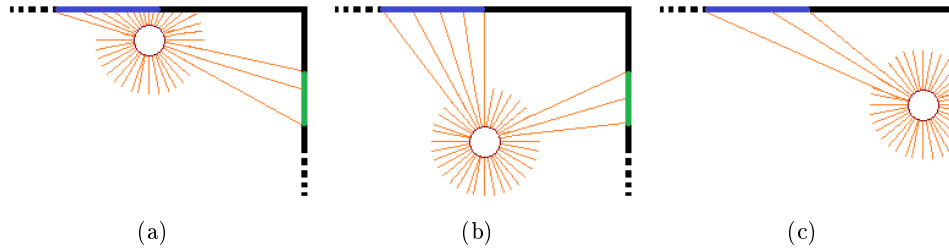


FIGURE 4.7 – Exemple d'un environnement (partiel) composé en particulier d'un objet bleu et d'un objet vert. Pour trois positions différentes du robot (modélisé par le cercle), les images (a), (b), et (c) représentent la proportion de  $\varphi$  qui inclue la projection de chaque objet. Clairement, ce nombre varie d'une image à l'autre. La somme des distances entre  $M(\varphi)$  voisins (extrémités des rayons partant du robot) est quant à elle stable pour un objet donné.

#### 4.4.4.4 Le noyau photométrique

A ce stade des développements, le noyau  $f$  permet 1) de sélectionner l'information perçue la plus pertinente ( $f_p$ ), et 2) de la pondérer en fonction de l'éloignement de la scène ( $f_s$ ). Reste à définir la façon dont chaque point  $\mathbf{x}(\theta, \varphi)$  impacte la signature finale, c'est-à-dire de quelle manière est utilisée l'information photométrique qui le définit. Par soucis de clarté, nous considérons pour l'instant  $\mathbf{x}$  comme un signal monochromatique, qui contient, pour chaque coordonnée, une valeur en niveau de gris codée sur 8 bits ( $\mathbf{x}(\theta, \varphi) \in [0, 255]$ ).

Nous définissons le noyau photométrique  $f_m$  en utilisant les noyaux monômes proposés par Siggelkow [Siggelkow 2002] et définis par :

$$f_m(\theta, \varphi) = \sqrt[\sum j]{\prod \mathbf{x}^j(\theta', \varphi')} \quad (4.24)$$

où  $\theta' = \theta + kd\theta$ ,  $\varphi' = \varphi + ld\varphi$  (avec  $k, l, j$  des entiers), et  $\sum j$  est l'ordre du noyau. Le plus simple d'entre eux (monôme d'ordre 1 avec  $k = 0, l = 0$ ),  $f_{m^1}(\theta, \varphi) = \mathbf{x}(\theta, \varphi)$ , n'est rien d'autre que la valeur du point étudié. Une intégrale de Haar fondée sur ce noyau correspond à la moyenne des niveaux de gris sur l'ensemble de la configuration  $\mathbf{x}$ . Cette solution ne prend pas en compte la structure de  $\mathbf{x}$  (l'information photométrique portée par le voisinage de chaque point). Ainsi, les trois images présentées Figure. 4.8 ont, pour un noyau  $f = f_{m^1}$ , une intégrale de Haar rigoureusement identique. Pourtant, elles sont clairement issues de trois lieux différents. Nous choisissons dans ce travail de marquer cette différence par l'utilisation d'un monôme d'ordre 2, défini à partir des couples  $(k = 0, l = 1)$  et  $(k = 2, l = 0)$ . Le noyau photométrique  $f_m$  ainsi construit s'écrit :

$$f_m(\theta, \varphi) = \sqrt{\mathbf{x}(\theta, \varphi + d\varphi) \cdot \mathbf{x}(\theta + 2d\theta, \varphi)} \quad (4.25)$$

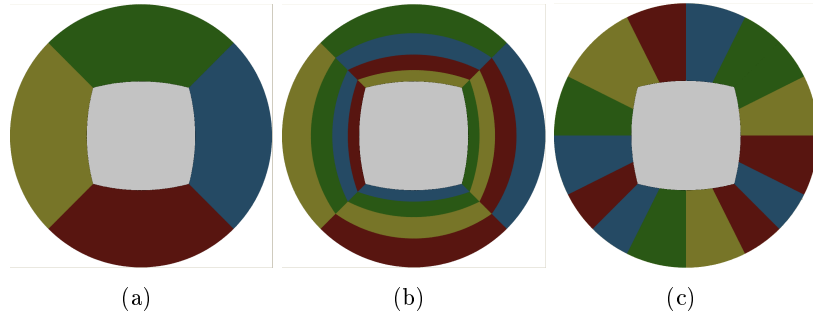


FIGURE 4.8 – Mise en évidence de l'importance du noyau photométrique. Les trois images présentent une information photométrique rigoureusement identique, mais répartie différemment. Sans le noyau photométrique, l'intégrale de Haar obtenue serait rigoureusement identique.

Un tel choix est justifié par plusieurs raisons. D'abord, il permet de tenir compte des voisins dans deux directions orthogonales. Ensuite, les faibles valeurs choisies pour  $k$  et  $l$  font que ce noyau est à support compact, une condition nécessaire comme expliqué précédemment. Enfin, le faible degré du monôme ( $j = 2$ ) garantit des performances computationnelles suffisantes pour une application temps réel.

#### 4.4.4.5 Le noyau de partitionnement

La dernière composante du noyau est élaborée afin de répondre au principal reproche généralement émis à l'encontre des invariants de Haar : leur faible pouvoir discriminant. Ils résument en effet une classe d'équivalence à un unique scalaire.

Dans un contexte de robotique mobile (ou d'indexation d'images en général), il existe donc un risque non négligeable que deux classes d'équivalence différentes produisent des signatures trop proches pour être distinguées. Comme expliqué plus tôt, nous proposons de réduire ce risque en construisant une famille d'intégrales obtenues à partir d'un ensemble de fonctions noyau au lieu d'une seule.

Nous les choisissons afin d'obtenir une signature sous forme d'histogramme, bénéficiant ainsi de l'ensemble des outils existants pour la manipulation de telles distributions. Ce résultat est obtenu en composant le noyau monôme  $f_m$  avec un ensemble de fonctions triangles  $f_{\Delta_i}$  qui partitionnent l'intervalle  $[0, 255]$  sous la forme d'un histogramme flou. Au contraire d'un histogramme classique (obtenu en utilisant des fonctions porte), ce type de distribution permet de rendre progressives les transitions entre partitions voisines (voir 4.9), donc de garantir une meilleure robustesse aux différents bruits liés au système dans son ensemble (capteur, interpolations, ...).

Soit  $n$  le nombre de partitions que l'on souhaite créer. La famille de fonctions triangle est définie par :

$$F_{\Delta}(v) = \{f_{\Delta_i}(v) = \max(0, 1 - \|\frac{n}{256}v - i\|), i = 0, \dots, n\} \quad (4.26)$$

où  $v$  est une valeur comprise entre 0 et 255.

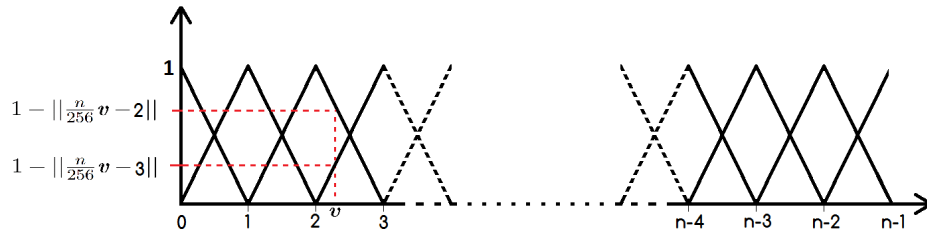


FIGURE 4.9 – Représentation de la classe de noyaux  $F_{\Delta}$ . Chaque triangle représente une fonction  $f_{\Delta_i}$  différente. Par exemple, pour la valeur  $v$  proposée, les valeurs de  $f_{\Delta_2}$  et  $f_{\Delta_3}$  sont représentées (pour toutes les autres fonctions triangle, la valeur est nulle). A noter que les fonctions triangles sont réparties sur l'ensemble de l'intervalle  $[0, 255]$  si bien que toutes les valeurs sont prises en compte par au moins l'une des fonctions  $f_{\Delta_i}$ .

#### 4.4.4.6 Combinaison des différents noyaux

Nous sommes maintenant en position de définir la fonction noyau dans son intégralité. Pour rappel, elle est composée :

- d'un noyau de pertinence qui permet de ne considérer que la portion utile de la sphère ;
- d'un noyau structurel qui pondère les objets de la scène par leur éloignement ;
- d'un noyau photométrique qui définit chaque point en fonction de ses voisins
- d'un noyau de partitionnement qui permet de construire une distribution d'invariants pour chaque configuration.

La famille  $F = f_i$ ,  $i = 0, \dots, n$  de noyaux construits s'exprime alors comme la composition du noyau photométrique par le noyau de partitionnement à laquelle sont multipliés les deux autres noyaux. On a :

$$f_i(\theta, \varphi) = f_p(\theta, \varphi) \cdot f_s(\theta, \varphi) \cdot (f_{\Delta_i} \circ f_m(\theta, \varphi)) \quad (4.27)$$

avec  $f_{\Delta_i} \in F_{\Delta}$

#### 4.4.5 Expression finale de la signature

A ce stade, l'ensemble des éléments nécessaires à l'élaboration de l'intégrale de Haar est défini. Certains ajustements sont toutefois nécessaires afin de pouvoir l'utiliser dans le contexte présent. D'abord, la configuration est définie sur une grille discrète et non dans un espace continu. Nous remplaçons donc le processus d'intégration par une sommation. Ensuite, elle contient pour chaque coordonnée un vecteur tridimensionnel  $\mathbf{x}(\theta, \varphi) = (\mathbf{x}^r(\theta, \varphi), \mathbf{x}^g(\theta, \varphi), \mathbf{x}^b(\theta, \varphi))^t$  et non un simple niveau de gris. L'intégrale est donc calculée indépendamment pour chaque canal  $R, G, B$ , produisant pour chaque famille de noyaux trois distributions au lieu d'une seule. Enfin, afin d'obtenir un histogramme normalisé relativement au contenu utile et non à l'ensemble de la sphère, le terme de normalisation est augmenté des noyaux de pertinence et structurels.

La signature  $Sig(I_t)$  de l'image  $I_t$  s'exprime finalement par :

$$Sig(I_t) = \left\{ \frac{\sum_{\theta=0}^{\pi} \sum_{\varphi=0}^{2\pi} f_p(\theta, \varphi) \cdot f_s(\theta, \varphi) \cdot f_{\Delta_i}(\sqrt{\mathbf{x}^c(\theta, \varphi + \partial\theta)} \cdot \mathbf{x}^c(\theta + 2\partial\theta, \varphi) \sin \theta \partial\theta \partial\varphi)}{\sum_{\theta=0}^{\pi} \sum_{\varphi=0}^{2\pi} f_p(\theta, \varphi) \cdot f_s(\theta, \varphi) \sin \theta \partial\theta \partial\varphi} \right\} \quad (4.28)$$

avec  $\partial\theta$  et  $\partial\varphi$  le pas d'échantillonnage de la sphère d'équivalence,  $i = 0, \dots, n$  l'indice de la partition étudiée parmi les  $n$  bins composant la famille de noyaux  $F_{\Delta}$ , et  $c = r, g, b$  le canal RGB étudié. La signature peut alors être représentée sous la forme de trois histogrammes normalisés (un par canal de couleur), comme illustré Figure.4.10

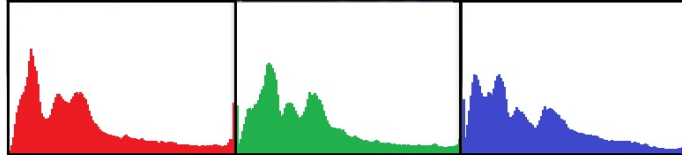


FIGURE 4.10 – Signature  $Sig(I_t)$  du lieu  $I_t$  présentée sous la forme d'une distribution par canal. Chaque bin (128 par canal) est le résultat du processus d'intégration pour une fonction de partition  $f_{\Delta_i}$ .

## 4.5 Calcul effectif, implémentation et optimisations

### 4.5.1 Analyse préalable : les différentes étapes

Nous étudions ici la complexité computationnelle de notre signature. En suivant l'équation 4.28, le calcul de  $Sig(I_t)$  nécessite à chaque image un ensemble d'étapes qui génèrent une quantité de calcul très élevée. Il faut :

- construire le masque de la portion de l'image qui représente les obstacles,
- projeter sur la sphère d'équivalence ledit masque, ainsi que l'image en elle-même et les frontières  $\Gamma$  de l'espace navigable,
- calculer le noyau photométrique pour chaque point utile de la sphère,
- déterminer pour chaque  $\varphi$  la projection dans le plan sol  $M(\varphi)$  du point  $m(\varphi)$  qui marque la frontière entre espace navigable et obstacles,
- et enfin, en utilisant tous ces éléments, évaluer l'équation 4.28 pour chaque canal  $R, G, B$  et chaque noyau  $f_{\Delta_i} \in F_{\Delta}$ .

### 4.5.2 Calculs réalisés hors ligne

Afin d'accélérer le processus de construction des signatures, les calculs qui dépendent uniquement des propriétés géométriques du capteur (et non du contenu photométrique de l'image), sont effectués lors de la phase d'initialisation. Les résultats sont alors mémorisés afin d'y avoir accès pendant l'exécution de la mission. Soient  $\partial\theta$  et  $\partial\varphi$  les pas choisis par l'utilisateur pour échantillonner la sphère unitaire. A partir de ces derniers, et étant donné les domaines de définition de  $\theta \in [0, \pi]$  et  $\varphi \in [0, 2\pi]$ , il est possible de déterminer l'ensemble des couples  $(\theta, \varphi)$  qui vont permettre de définir la configuration  $\mathbf{x}$ . La construction de la signature implique de connaître pour chacun d'entre eux 1) le pixel correspondant dans l'image, et 2) les coordonnées du point 3d s'y projetant si ce dernier appartient au sol. Ces informations ne dépendent pas du contenu de l'image, et peuvent donc être calculées en amont. Nous définissons deux tables de correspondances  $\mathcal{L}_{\mathcal{I}}$  et  $\mathcal{L}_{\mathcal{W}}$  qui vont stocker pour chaque couple  $(\theta, \varphi)$  les coordonnées de leur projection, respectivement dans l'image et dans le plan du sol 3D (lorsqu'elle existe). Pour cela, nous supposons connus :

- les paramètres liés au système d'acquisition (la matrice  $\mathbf{K}$  et le paramètre  $\xi$  du miroir) ;
- la hauteur  $H$  du centre optique de la caméra relativement au sol ;
- le rayon  $R$  du cercle modélisant la projection du miroir dans l'image.

Le capteur est donc supposé calibré. En utilisant les équations (4.12) et (4.22), les deux tables de correspondance peuvent être construites (voir Algorithme.6). De cette façon, les nombreux calculs trigonométriques qui auraient dû être réalisés pour chaque image sont remplacés par une simple lecture dans une matrice, ce qui, d'un point de vue computationnel, est beaucoup plus performant.

**Algorithme 6** Construction de  $\mathcal{L}_{\mathcal{I}}$  et  $\mathcal{L}_{\mathcal{W}}$ .

**Entrées :** Paramètres de calibration  $(K, H, \xi, R)$   
 Pas d'échantillonnage  $\partial\theta$  et  $\partial\varphi$  de la grille sphérique  
**Sorties :** Tables de correspondance  $\mathcal{L}_{\mathcal{I}}$  et  $\mathcal{L}_{\mathcal{W}}$

$n_\theta = \frac{\pi}{\partial\theta}$ ,  $n_\varphi = \frac{2\pi}{\partial\varphi}$  //  $n_\theta$  et  $n_\varphi$  sont des entiers

**Pour tout**  $i = 0, \dots, n_\theta$  **Faire**

**Pour tout**  $j = 0, \dots, n_\varphi$  **Faire**

$\theta = i\Delta\theta$ ,  $\varphi = j\Delta\varphi$

*Construction de  $\mathcal{L}_{\mathcal{I}}$  :*

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ 1 \end{pmatrix} = \frac{\sin\theta}{\cos\theta + \xi} \cdot \mathbf{K} \cdot \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 1 \end{pmatrix}$$

**Si**  $(u - u_0)^2 + (v - v_0)^2 > R^2$  **Alors**  $\mathcal{L}_{\mathcal{I}}(i, j) = \emptyset$

**Sinon**  $\mathcal{L}_{\mathcal{I}}(i, j) = (u, v)$

**Fin Si**

*Construction de  $\mathcal{L}_{\mathcal{W}}$  :*

**Si**  $\theta > \pi/2$  **Alors**

$X = H \cos\varphi \tan\theta$

$Y = H \sin\varphi \tan\theta$

$\mathcal{L}_{\mathcal{W}}(i, j) = (X, Y)$

**Sinon**

$\mathcal{L}_{\mathcal{W}}(i, j) = \emptyset$

**Fin si**

**Fin pour**

**Fin pour**

### 4.5.3 Optimisation des calculs en ligne

L'autre partie des calculs est directement liée au contenu de l'image  $I_t$  acquise à l'instant  $t$ , et doit donc être effectuée en ligne. Elle comprend :

- la sélection des points  $(\theta, \varphi)$  pertinents par rapport à l'espace libre extrait ;
- l'évaluation des distances  $\|\mathbf{M}(\varphi) - \mathbf{M}(\varphi \pm \partial\varphi)\|$  permettant de construire  $f_s$  ;
- le calcul du noyau monomial  $f_m$  ;
- la sommation finale pour chaque canal  $R, G, B$  et chaque fonction noyau  $f_{\Delta_i} \in F_{\Delta}$ .

L'ensemble de ces opérations est réalisé en seulement deux parcours du plan  $\Pi(\theta, \varphi)$  : un premier qui permet de localiser pour chaque  $\varphi$  le point frontière sol/obstacles  $\mathbf{m}(\varphi)$ , et un second qui exploite cette information pour construire la signature dans son intégralité (tous les noyaux et tous les canaux en même temps).

Pour le premier, nous supposons connu le masque  $\mathcal{M} = \mathbf{FS} \cup \overline{\mathbf{FS}}$  (dans l'image omnidirectionnelle) qui labellise chaque pixel comme appartenant à l'espace libre ou à son complément. L'idée est de parcourir chaque  $\varphi$  à la recherche du premier  $\theta$  tel

que  $f_{img}(\theta, \varphi) \in \mathbf{FS}$ . Nous utilisons pour cela la table de correspondance  $\mathcal{L}_{\mathcal{I}}$  qui, on le rappelle, a été construite afin que  $f_{img}(\theta, \varphi) = I_t(\mathcal{L}_{\mathcal{I}}(\frac{\theta}{\partial\theta}, \frac{\varphi}{\partial\varphi}))$ .

Le second parcours utilise cette information d'une part pour ne considérer que la portion utile de la sphère, et d'autre part pour calculer le noyau structurel qui est commun à tous les points d'un même  $\varphi$ . En outre, dans la pratique, tel que représenté Fig.4.9, chaque point de la sphère n'impacte au final qu'au plus deux  $f_i$  par canal  $R, G, B$ . Plutôt que de parcourir le spectre complet de ces fonctions, l'idée est donc de localiser celles qui sont impactées, et ainsi, considérer pour un point donné uniquement les intégrales pour lesquelles sa contribution est non nulle.

La structure de l'algorithme réalisant ces deux parcours, ainsi que le détail des calculs qui y sont effectués sont présentés Algorithme.7.

## 4.6 Mesure de distance entre signatures

A travers la signature proposée, le robot doit pouvoir déterminer si la scène qu'il observe est un lieu préalablement visité. Pour cela, il est nécessaire d'estimer la ressemblance entre deux signatures. Nous avons volontairement pris le parti de construire une signature se présentant sous la forme d'une distribution, afin d'avoir à disposition l'ensemble des outils qui existent pour la manipulation d'histogrammes.

Soient  $Sig(I_1) = (Sig^r(I_1), Sig^g(I_1), Sig^b(I_1))$  et  $Sig(I_2) = (Sig^r(I_2), Sig^g(I_2), Sig^b(I_2))$  deux signatures se présentant chacune sous la forme de trois distributions indépendantes. Nous proposons de mesurer la distance entre  $Sig(I_1)$  et  $Sig(I_2)$  à partir de la moyenne des distances entre chaque canal  $R, G, B$  :

$$d(Sig(I_1), Sig(I_2)) = \frac{\sum_{c=r,g,b} d(Sig^c(I_1), Sig^c(I_2))}{3} \quad (4.29)$$

Il existe de nombreuses distances entre distributions, traditionnellement réparties en deux grandes familles :

- les distances terme à terme comme les normes  $L_1, L_2, L_\infty$ , la distance de Jeffrey ou encore l'intersection d'histogrammes. Ces distances comparent les partitions de même indice puis concatènent les résultats obtenus sur l'ensemble de la distribution (somme, moyenne, ...)
- les distances entre distributions considérées comme un tout (*EMD*, distance quadratique, mesure de Kolmogorov-Smirnov, ...). Elles sont généralement plus lentes à calculer, et élaborées pour être plus tolérantes au bruit agissant sur chaque distribution.

Dans ses travaux de thèse, Charron [Charron 2006] utilise entre autres l'EMD (*Earth Mover Distance*), qui, d'après Rubner [Rubner 2000], présente les meilleurs résultats. Néanmoins, cette solution est extrêmement coûteuse en temps de calcul, puisqu'elle implique de passer par une phase de minimisation de complexité  $O(n^2)$ , où  $n$  est le nombre de partitions. Dans une application temps réel, c'est clairement inenvisageable, la signature construite à un instant  $t$  devant être comparée à



**Algorithme 7** Construction en ligne de la signature  $Sig(I_t)$ 

**Entrées :** Image omnidirectionnelle  $I_t$ , masque binaire  $\mathcal{M} = \mathbf{FS} \cup \overline{\mathbf{FS}}$

**Sortie :** Signature  $Sig(I_t)$

$$n_\theta = \frac{\pi}{\partial\theta}, \quad n_\varphi = \frac{2\pi}{\partial\varphi}$$

**Premier parcours : recherche des  $m(\varphi)$**

**Pour tout**  $j = 0, \dots, n_\varphi$  **Faire**

$i = n_\theta/2$  // Le sol ne peut être projeté au dessus de l'équateur de la sphère

**Faire**

$$i = i + \Delta\theta$$

**Tant que**  $I_t(\mathcal{L}_I(i, j)) \notin \mathbf{FS}$  // C'est-à-dire t.q.  $\mathcal{M}(\mathcal{L}_I(i, j)) = \overline{\mathbf{FS}}$

$$\theta_{Max}(j) = i$$

**Fin pour**

**Second parcours : construction de la signature**

$$Somme^R = 0, \quad Somme^G = 0, \quad Somme^B = 0$$

**Pour tout**  $j = 0, \dots, n_\varphi$  **Faire**

$$f_s = \frac{1}{2} \cdot (d(\mathcal{L}_W(\theta_{Max}(j), j), \mathcal{L}_W(\theta_{Max}(j-1), j-1)) + d(\mathcal{L}_W(\theta_{Max}(j), j), \mathcal{L}_W(\theta_{Max}(j+1), j+1)))$$

$$i = \theta_{Max}(j)$$

**Faire**

**Pour**  $c = R, G, B$  **Faire**

$$f_m^c = \sqrt{I_t^c(\mathcal{L}_I(i, j+1))I_t^c(\mathcal{L}_I(i+2, j))}$$

$$b^- = \lfloor \frac{f_m^c}{n_b} \rfloor$$

$$b^+ = \lceil \frac{f_m^c}{n_b} \rceil$$

$$Sig(I_t)_{b^-}^c = f_s \cdot (1 - \|\frac{n_b}{256} f_m^c - b^-\|) \cdot \sin \theta$$

$$Sig(I_t)_{b^+}^c = f_s \cdot (1 - \|\frac{n_b}{256} f_m^c - b^+\|) \cdot \sin \theta$$

$$Somme^c = f_s$$

**Fin pour**

$$i = i + 1$$

**Tant que**  $\mathcal{L}_I(i, j) \neq \emptyset$

**Fin pour**

**Phase de normalisation**

**Pour chaque**  $i = 0, \dots, n_b - 1, c = R, G, B$  **Faire**

$$Sig(I_t)_i^c = \frac{S(I_t)_i^c}{Somme^c}$$

celles caractérisant les lieux déjà existants. Notre choix s'est donc tourné vers l'intersection d'histogramme, qui est une mesure de distance bin à bin, ayant montré dans [Barla 2003] de bonnes performances pour la classification d'images couleur. L'inconvénient d'une telle approche (pas de robustesse face à un simple décalage de l'information) est en partie contrebalancé par le fait que nos distributions sont construites à partir de fonctions triangles, qui lissent les transitions entre partitions voisines. Elle est en tout cas très rapide à calculer, et ne ralentit donc pas le système lorsqu'il doit se localiser dans son environnement.

Sachant que les distributions sont normalisées, la distance entre deux signatures est alors définie par :

$$d(\text{Sig}(I_1), \text{Sig}(I_2)) = \frac{\sum_{c=r,g,b} \sum_{i=0}^n \min(\text{Sig}(I_1)^c(i), \text{Sig}(I_2)^c(i))}{3} \quad (4.30)$$

Le résultat se présente sous la forme d'un pourcentage de ressemblance compris entre 0 et 1. Une valeur de 1 indiquera une ressemblance absolue tandis qu'une valeur de 0 sous entendra deux distributions qui n'ont strictement rien en commun.

## 4.7 Validation expérimentale

### 4.7.1 Avant-propos

La signature a été développée afin d'être utilisée dans un processus d'indexation de lieu, pour fournir au robot mobile un moyen de se localiser en environnement intérieur ou extérieur. La validation expérimentale s'articule donc autour de deux scénarios aux propriétés différentes :

- un scénario en intérieur de taille modeste, qui vise à mettre en évidence la plus-value en terme d'invariance de notre approche, relativement à un simple histogramme couleur ;
- un scénario en extérieur de taille plus importante, qui quant à lui montre son apport en terme de discriminance.

Nous profitons également de ces deux scénarios pour aborder la question de la cartographie qui sera au coeur du prochain chapitre.

### 4.7.2 Scénario en intérieur

#### 4.7.2.1 Présentation du scénario

Notre premier scénario se déroule dans un environnement en intérieur composé de 3 pièces, dont la géométrie est présentée Fig.4.11(a). Bien que simple par sa taille, il présente un vrai challenge d'un point de vue photométrique. Comme illustré figure 4.11(b), deux des trois pièces sont visuellement très difficiles à distinguer, y compris pour un observateur humain. En outre, la plus grande des pièces présente des conditions d'éclairage différentes suivant la position qu'y occupe le robot (voir figure 4.11(c)).

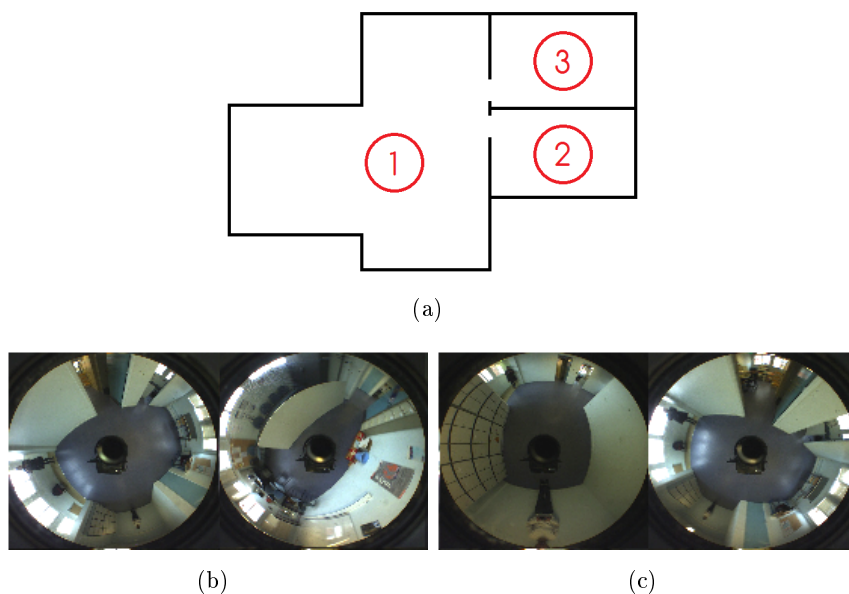


FIGURE 4.11 – (a) Disposition des trois pièces formant l’environnement du scénario. (b) Mise en évidence de la similarité photométrique entre les pièces 1 et 3. (c) Mise en évidence des différences d’éclairage au sein de la pièce 1.

Dans ce scénario, le robot est téléguidé et effectue plusieurs passages dans chacune des pièces, en réalisant l’acquisition d’une nouvelle image à une fréquence de  $10Hz$ . Au total, plus de 3200 images sont acquises. Nous proposons de les utiliser pour générer une analyse à la fois quantitative et qualitative de la signature.

#### 4.7.2.2 Évaluation quantitative

Quinze images sont sélectionnées et labellisées à la main en fonction de leur appartenance à l’une ou l’autre des pièces. Leur position géographique est judicieusement choisie, afin de correspondre à un panel le plus représentatif possible de chacune d’entre elles. Ces différentes images de référence, ainsi que leurs positions respectives sont présentées Figure 4.12.

L’idée est alors de mesurer la distance entre chaque couple d’images afin d’évaluer la capacité de notre signature à distinguer un positionnement dans l’une ou l’autre des pièces. Dans [Deselaers 2008], une étude comparative des différentes signatures (locales comme globales) utilisées en reconnaissance d’images est proposée. Elle met en évidence le bon comportement des histogrammes couleur relativement aux autres approches. Couplé au fait qu’il s’agit d’une distribution et que la même mesure de distance peut par conséquent y être appliquée, nous proposons de nous en servir comme base de comparaison. Les distances entre chaque couple d’images obtenues par l’intermédiaire des deux signatures sont présentées sous forme de matrices Figure 4.13. Les portions colorées désignent les couples d’images appartenant à une même pièce. Il s’agit donc de celles censées présenter la ressemblance la plus

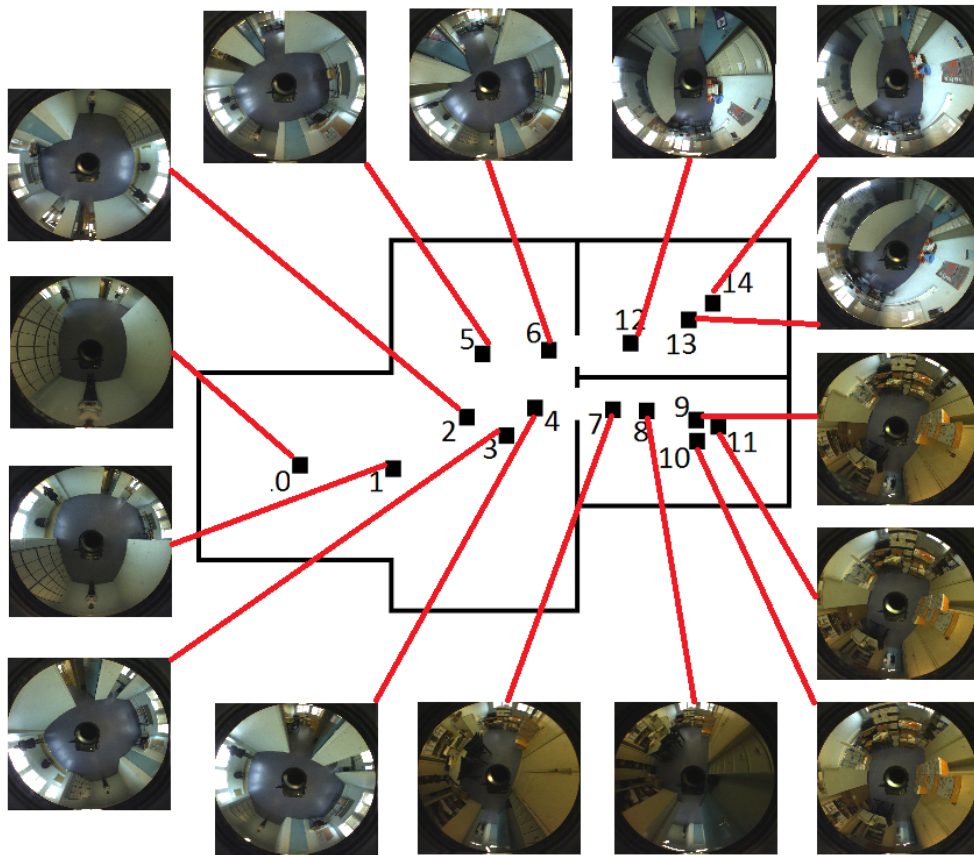


FIGURE 4.12 – Images représentatives de chaque lieu, ainsi que leur position dans l’environnement.

forte. Nous portons en particulier l’attention du lecteur sur deux régions précises de la matrice (mises en évidence par un grossissement).

La première région correspond aux distances calculées entre l’image acquise à la position 0 et toutes les autres. L’image 0 appartenant à la première pièce, sa signature devrait être proche de celles des images 1 à 6, et la plus éloignée possible des autres. En utilisant l’histogramme couleur, ce n’est clairement pas le cas : hormis la position 1 qui présente une ressemblance de 78%, toutes les autres positions dans la pièce 1 présentent une mesure de similarité allant de 0.62 à 0.64. Pourtant, lorsque la signature de cette même image 0 est comparée à celles acquise dans la pièce 2, les valeurs vont de 0.72 à 0.77. Autrement dit, la ressemblance est plus forte, alors que géographiquement, les lieux sont différents : il y a ambiguïté perceptive. Au contraire, en utilisant notre approche fondée sur les invariants de Haar, l’intersection entre les signatures issues de la première pièce est toujours supérieure à 0.84 tandis que lorsque les images proviennent de deux lieux différents, la mesure est au plus de 0.7. La séparation est donc beaucoup plus nette.

Le second grossissement présente les distances mesurées entre les images issues

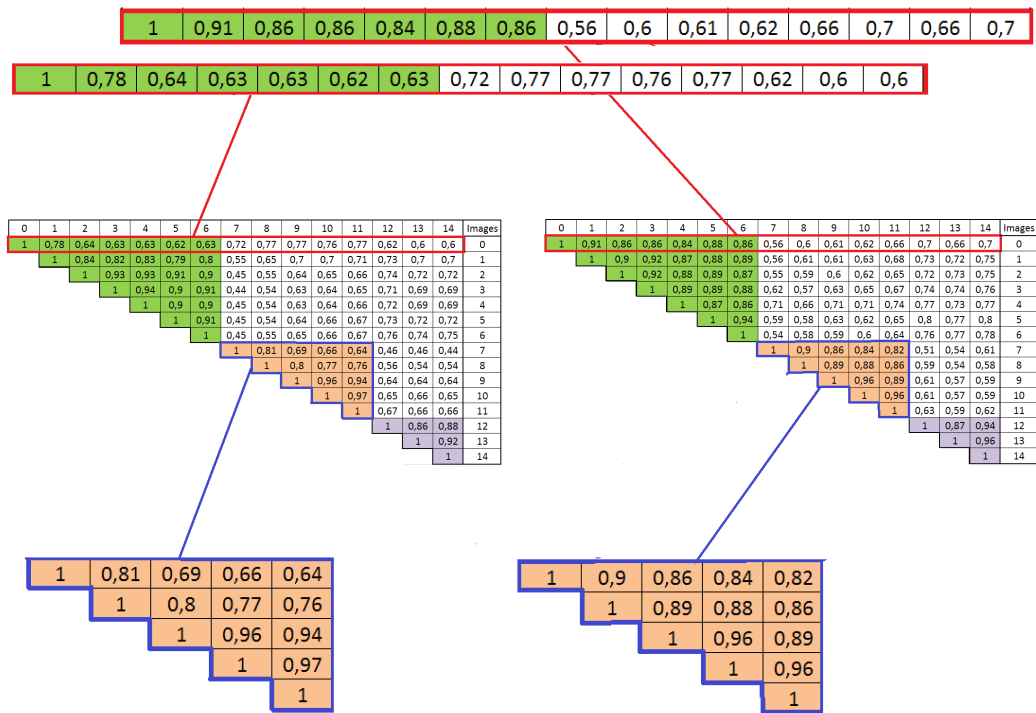


FIGURE 4.13 – Matrices de similarités présentant la distance entre chaque couple d’images en utilisant **à gauche** l’histogramme couleur, et **à droite** la signature proposée.

de la deuxième pièce. Encore une fois, on constate un net avantage de la signature proposée relativement à l’histogramme couleur. En effet, les distances mesurées dans le cas des histogrammes couleur oscillent entre 0,64 et 0,97, ce qui remet en question la capacité d’invariance d’une telle représentation. Notre méthode quant à elle est beaucoup plus stable puisque les valeurs s’échelonnent entre 0,82 et 0,96, rendant beaucoup plus probable une bonne localisation du robot.

Pour donner un sens quantitatif à ces observations, nous proposons d’utiliser le critère de Fisher qui mesure la séparabilité entre classes. Il est défini par :

$$\mathcal{F} = (\nu_1 - \nu_2)^2 / (\sigma_1^2 + \sigma_2^2) \quad (4.31)$$

où  $\nu_1$  et  $\nu_2$  désignent les centroïdes des deux classes étudiées, et  $\sigma_1^2, \sigma_2^2$  leur variance respective. La valeur obtenue est d’autant plus élevée que les deux classes sont différentes. Appliqué à notre contexte, nous avons donc calculé cette valeur pour chaque couple de pièces, et présenté les résultats dans le tableau 4.1. Les résultats obtenus confirment l’affirmation précédente, puisque la distinction entre lieux est clairement plus forte en utilisant notre signature qu’avec l’histogramme couleur.

critère de Fisher :	$\mathcal{F}(C_1, C_2)$	$\mathcal{F}(C_1, C_3)$	$\mathcal{F}(C_2, C_3)$
Histogramme couleur	2.88	2.48	4.32
Signature proposée	6.28	3.51	12.83

TABLE 4.1 – Résultats comparatifs entre l’histogramme couleur et la signature proposée fondée sur le critère de Fisher. La classe  $C_i$  est construite à partir de l’ensemble des images acquises dans la pièce  $i$ .

### 4.7.2.3 Analyse qualitative

En complément de cette étude, nous nous sommes également intéressés à la capacité du robot à définir de manière autonome des lieux dans cet environnement, et à s’y localiser chaque fois que sa position correspond. Il s’agit en fait d’une forme de cartographie simplifiée qui est une bonne introduction au chapitre à venir. La procédure passe par l’utilisation de deux seuils  $\tau_1 = 0.9$ , et  $\tau_2 = 0.85$ , qui sont utilisés comme suit :

- A l’instant initial, le robot génère un lieu  $\mathcal{L}_0$  qu’il définit à partir de la signature  $Sig(I_0)$ .
- A chaque nouvelle image  $I_t$ , la signature  $Sig(I_t)$  construite est comparée avec la signature  $S(\mathcal{L}_0)$  du lieu  $\mathcal{L}_0$ . Dès que la distance  $d(Sig(I_t), Sig(\mathcal{L}_0))$  est inférieure au seuil  $\tau_1$ , le robot génère un nouveau lieu  $\mathcal{L}_1$  qu’il caractérise à partir de  $Sig(I_t)$ .
- Le seuil  $\tau_2$  sert de son côté à détecter une fermeture de boucle, c’est-à-dire le passage d’un lieu courant  $\mathcal{L}_t$  à un lieu  $\mathcal{L}_i$  déjà visité par le robot : lorsque la ressemblance entre l’image courante et le lieu courant est inférieure au seuil  $\tau_1$  (changement de lieu), le robot cherche parmi les autres lieux mémorisés s’il en existe un dont la similarité est supérieure au seuil  $\tau_2$ . Si tel est le cas, il estime l’avoir atteint. Sinon, il en génère un nouveau.

Nous avons appliqué cette procédure sur l’ensemble de la séquence, et montré le résultat sur la figure 4.14. Chaque point y apparaissant correspond à une position dans l’environnement ayant mené à la création d’un nouveau lieu, tandis que les segments les reliant représentent les relations d’adjacence qui existent entre eux. Clairement, les liens entre les différents lieux sont cohérents avec la réalité physique des déplacements du robot. Tout au long des 3200 images, le robot a su détecter d’une part, lorsqu’il ne se trouvait dans aucun lieu connu (en définissant alors un nouveau), et d’autre part, les fermetures de boucle, c’est-à-dire le passage dans un lieu préalablement visité. Il existe tout de même une exception : dans la pièce 3, une relation d’adjacence lie deux lieux qui ne sont pas géographiquement voisins. Lors d’un de ses passages dans la pièce, le robot a donc estimé se déplacer directement du lieu vert au lieu cyan, sans passer par le lieu magenta. Nous justifions ce comportement par la proximité photométrique entre les lieux vert et magenta : lorsqu’il a jugé avoir quitté le lieu vert ( $d(Sig(I_t), Sig(\mathcal{L}_t)) < \tau_1$ ), il avait déjà atteint une position plus proche du cyan que du magenta. Ce comportement est donc dû à l’effet d’hystérésis du double seuillage utilisé.

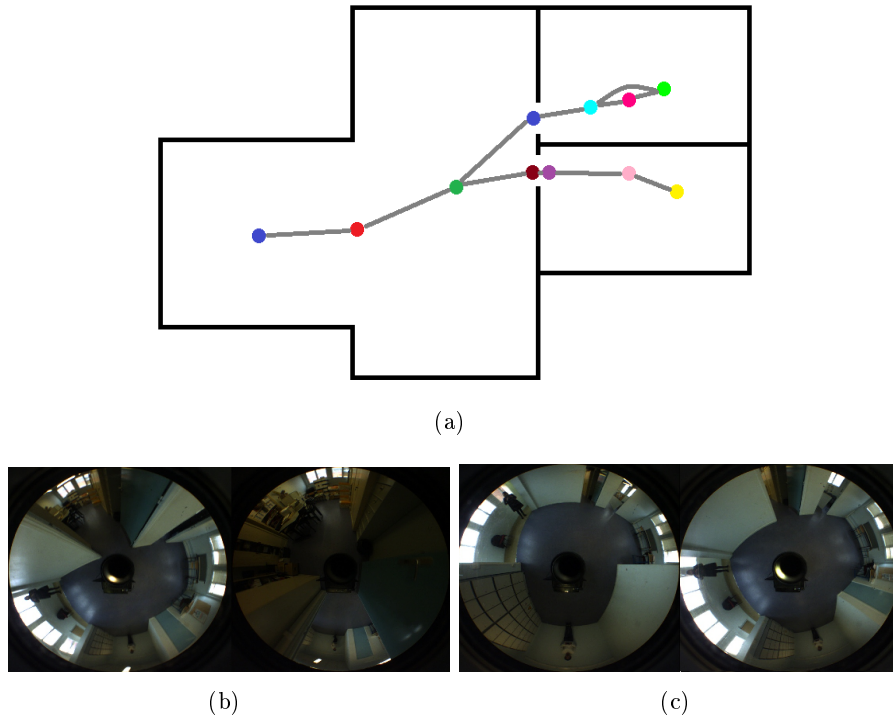


FIGURE 4.14 – (a) Localisation des différents lieux générés par le robot pour le scénario "3 pièces", ainsi que les relations d'adjacence qui existent entre eux. (b) et (c) : Illustration de l'absence de corrélation entre la distance au sol et la distance dans l'espace des signatures. (a) Les lieux "marron" et "mauve" sont géographiquement très proches, et pourtant très différents au sens photométrique. (b) Les lieux "rouge" et "vert" sont distants l'un de l'autre, et pourtant photométriquement voisins.

La position des différents lieux mérite également quelques commentaires. En particulier, ceux modélisés par les points marron et mauve sont géographiquement très proches. On constate toutefois qu'ils sont chacun situés d'un côté d'un encadrement de porte. En toute logique, lorsque le robot passe de l'un à l'autre, sa perception change très rapidement (voir Figure.4.14(b)), ce qui implique un déplacement important dans l'espace des signatures, et ce même s'il correspond à un mouvement de faible amplitude dans l'environnement. A l'inverse, les lieux rouge et vert sont séparés d'une distance assez conséquente, qui s'explique par une variation lente de la perception entre les deux lieux (notamment parce qu'ils appartiennent à la même pièce) comme on peut le constater Figure.4.14(c).

Quoi qu'il en soit, à aucun moment le robot ne s'est localisé dans la mauvaise pièce, ce qui nous conforte dans la capacité de notre signature à être utilisée comme index de lieu.

### 4.7.3 Scénario en extérieur

#### 4.7.3.1 Présentation du scénario

Pour ce second scénario, nous avons placé le robot en environnement extérieur urbain, et l'avons doté d'un module de navigation réactive fondé sur le squelette de l'espace navigable. Le robot se déplace cette fois de manière totalement autonome, en tenant sa droite chaque fois qu'il atteint une intersection. Au total, il suit une trajectoire d'environ 350m qui est représentée figure.4.15.

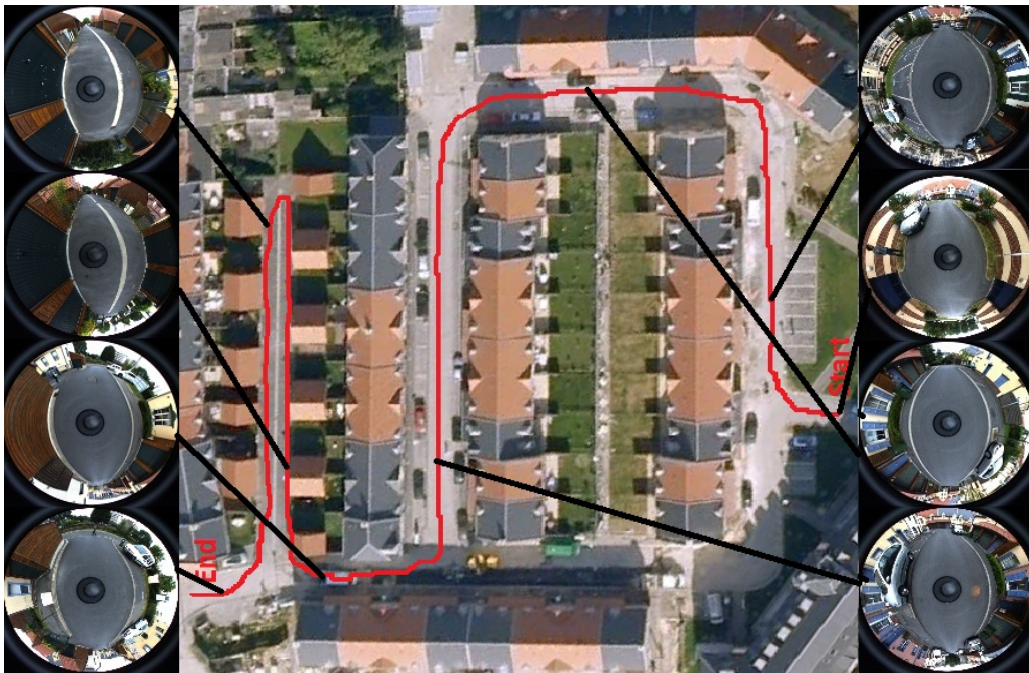


FIGURE 4.15 – Vue Google Earth du scénario en extérieur, assortie d'un ensemble de vues omnidirectionnelles de différents lieux concernés. En *rouge* : trajectoire (approximative) empruntée par le robot.

Nous identifions deux difficultés dans cette expérimentation :

- Ses dimensions, qui impliquent la génération d'un nombre de lieux important, donc augmente les risques de mauvaise localisation (donc de fermeture de boucle erronée).
- Les propriétés photométriques de certains lieux visités : l'environnement considéré est un quartier où les habitations se ressemblent beaucoup (voir Figure.4.16). Au fil de ses déplacements, le robot rencontre donc des situations géographiquement éloignées, mais photométriquement voisines, ce qui génère des risques importants d'aliasing perceptif.

L'ensemble des traitements est cette fois réalisé directement en ligne par le robot. Equipé d'un capteur catadioptrique de résolution 800 pixels par 600, associé à un processeur Intel I7-2620m, il lui faut un peu moins de 200ms pour définir l'espace



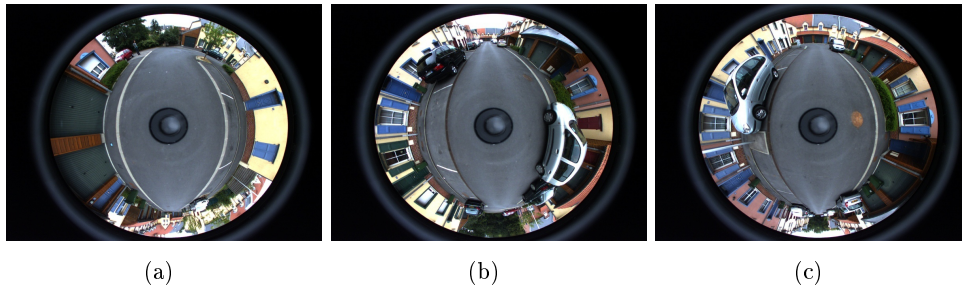


FIGURE 4.16 – Mise en évidence de la ressemblance photométrique entre différents lieux de l’environnement considéré. Globalement, bien que répartie différemment dans l’image, la distribution photométrique est très proche.

navigable, extraire le squelette dans l’image, calculer la signature photométrique, se localiser et enfin déterminer la commande de déplacement. Pour ne pas excéder sa capacité de traitements, il réalise donc l’acquisition d’une nouvelle image à une fréquence d’environ 5Hz. De plus, afin de rendre le processus de localisation plus robuste (notamment aux erreurs de segmentation du sol), ce dernier n’est réalisé qu’une fois par seconde, à partir de la moyenne des 5 signatures construites. Le robot se déplace à une vitesse d’environ 30cm/sec, ce qui rend cette cadence suffisante. La stratégie à deux seuils présentée pour le scénario en intérieur (avec  $\tau_1 = 0.9$ , et  $\tau_2 = 0.8$ ) est une nouvelle fois utilisée.

#### 4.7.3.2 Analyse des résultats

La représentation de l’environnement ainsi obtenue est modélisée figure 4.17. Elle est formée d’un total de 54 lieux qui sont liés par des relations d’adjacence cohérentes avec la trajectoire empruntée par le robot. Comme attendu, nous observons une nouvelle fois que la répartition des lieux sur la carte n’est pas homogène, ce qui illustre bien la notion d’orbite utilisée dans ce chapitre : tant que l’information visuelle reste inchangée (ou presque), la configuration construite appartient à la même classe d’équivalence que celle ayant mené à la génération du lieu courant. La distance parcourue par le robot n’a donc qu’un impact indirect dans la décision d’ajouter un nouveau lieu.

Au vu des résultats, la signature construite semble, au moins pour ce scénario, robuste à la question de l’aliasing perceptif. Afin de mettre un peu plus en évidence cette performance, nous avons rejoué la séquence d’images en appliquant la même procédure de génération de lieu, mais en omettant volontairement certaines composantes du noyau développé plus tôt. Nous nous intéressons donc aux conséquences sur les performances du système 1) lorsque le noyau structurel est ignoré (c’est-à-dire lorsque les points ne sont plus pondérés par leur distance au robot), et 2), lorsqu’en plus l’espace navigable est pris en compte dans le processus d’intégration.

Afin d’observer efficacement les changements impliqués par la modification du noyau, nous organisons les résultats sous la forme d’un graphe de correspondances

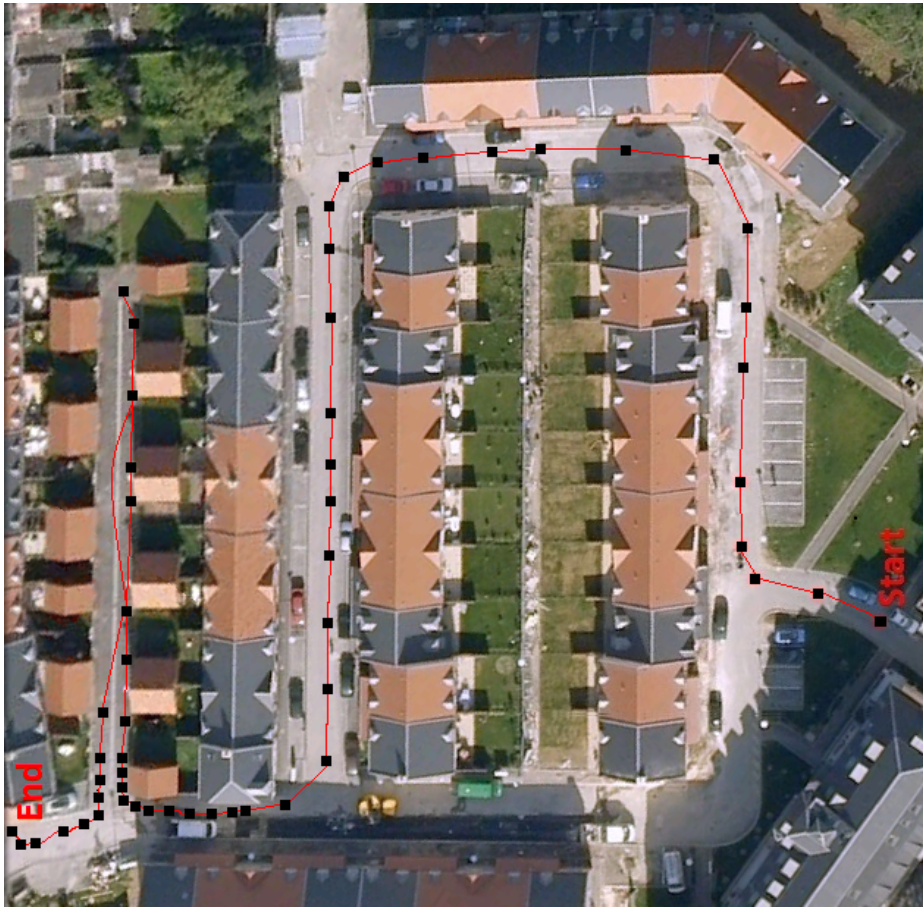
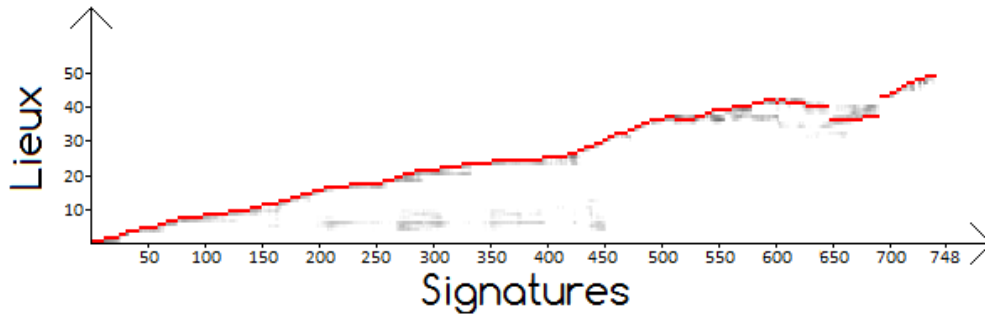


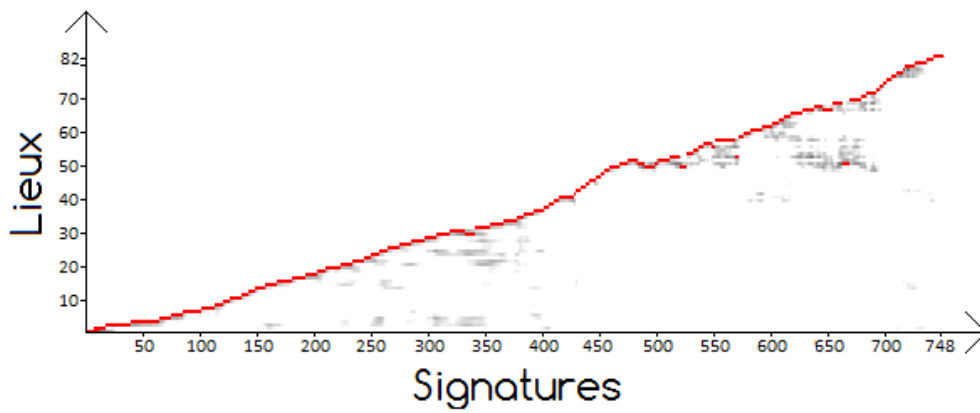
FIGURE 4.17 – Localisation dans l’environnement, des différents lieux (en noir) générés par le robot au cours de son exploration, ainsi que les relations d’adjacence qui les lient (en rouge).

(pour chaque noyau), où chaque signature construite (en abscisse) est associée au lieu sur lequel le robot estime se trouver (en ordonnée). Ces graphiques sont présentés Figure.4.18.

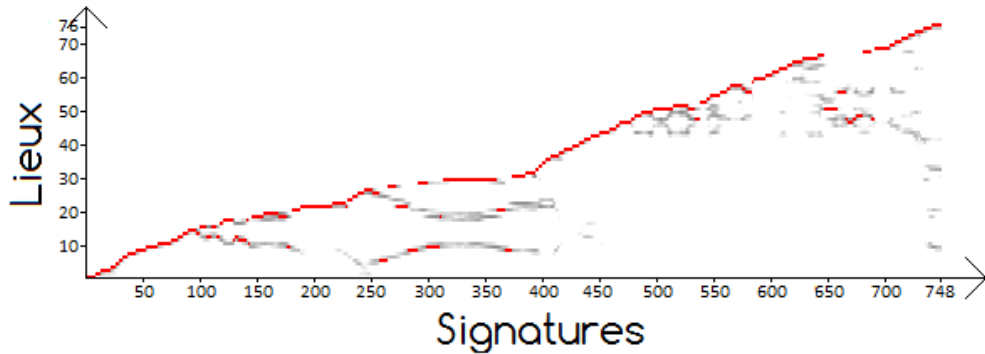
Commençons par comparer la signature proposée (Figure.4.18(a)), avec celle qui ignore le noyau structurel (Figure.4.18(b)). Globalement, l’aspect des deux courbes est similaire, mais deux différences importantes sont tout de même à souligner. La première réside dans le nombre de lieux nécessaires au robot pour caractériser l’environnement. Pour une trajectoire identique, 82 lieux sont générés par le robot lorsqu’il ignore le noyau structurel, contre seulement 54 lorsqu’il en tient compte. Cette différence est principalement due à la trajectoire du robot, qui passe à plusieurs reprises à distance réduite d’objets de l’environnement (murs, voitures). Comme nous l’avons vu au cours du chapitre, un faible déplacement du robot entraîne dans ces situations une modification importante de la perception desdits objets. C’est exactement dans le but de compenser ce phénomène que le noyau structurel a été introduit. La seconde différence est l’incapacité du robot à identifier la fermeture de



(a)



(b)



(c)

FIGURE 4.18 – Graphe de correspondances entre lieux et signatures. En *rouge*, la position de chaque signature pour : (a) la signature proposée complète, (b) la signature proposée sans le noyau structural, et (c) la signature proposée sans le noyau structural ni le noyau de pertinence. En *gris*, nous notifiions également pour chaque signature, les lieux dont la similarité, bien qu'inférieure à celle du lieu choisi, est suffisamment élevée pour être signalée (plus le pixel est sombre, plus la ressemblance est marquée).

boucle présente à la fin du scénario (voir Figure.4.17), lorsqu'il ne tient pas compte du noyau structurel. Il s'agit en fait d'une situation un peu particulière, puisque la voie navigable parcourue dans les deux sens par le robot, est coupée en son milieu par une ligne blanche continue. Cette dernière étant considérée comme un obstacle à part entière par le processus d'extraction de l'espace navigable, le robot emprunte des chemins différents à l'aller et au retour. L'information perçue varie alors suffisamment pour que la fermeture de boucle ne soit pas détectée, à moins que le noyau structurel ne vienne compenser cette différence.

Lorsque l'espace navigable est en plus ajouté au processus d'intégration, les résultats s'effondrent de manière encore plus évidente (voir Figure.4.18(c)). Les relations d'adjacence ne sont en effet plus fidèles aux déplacements du robot. Non seulement la fermeture de boucle n'est pas détectée, mais en plus de fausses fermetures de boucle le sont. Même en faisant varier les seuils  $\tau_1$  et  $\tau_2$ , nous n'avons jamais été en mesure de proposer une représentation de l'environnement cohérente avec sa réalité physique. La raison est simple : avec un capteur fixé à approximativement 1m35 au dessus du sol, l'espace navigable occupe approximativement un tiers de l'image. De ce fait, des variations mineures dans son aspect ou sa taille dominent les changements qui surviennent dans la scène (apparition d'un nouvel obstacle par exemple). La signature devient instable, et génère des erreurs de localisation.

## 4.8 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la question de l'indexation de lieu par construction de signature visuelle. Après un tour d'horizon des différentes approches existantes, nous avons proposé une méthode originale fondée sur le formalisme des invariants intégraux de Haar. Chaque élément lié à ce formalisme a été explicité, puis défini en tenant compte du contexte d'exploration en environnement inconnu. Ainsi :

- la géométrie du capteur est intégrée dans le processus de calcul d'invariant, ce dernier étant réalisé sur la sphère d'équivalence ;
- seule la portion de l'image correspondant au complément de l'espace libre est considérée, ce dernier ne contenant que peu d'information photométrique pertinente sur le lieu observé ;
- chaque pixel est pondéré par sa distance au robot, dans un soucis de compenser les variations de proportions liées aux déplacements du robot.

Afin de valider la capacité d'une telle signature à servir de critère dans un processus de localisation, nous l'avons sollicité dans deux scénarios aux propriétés et objectifs différents. Le premier, de taille modeste en environnement intérieur, a permis de mettre en évidence sa stabilité dans le temps, et surtout l'augmentation de son pouvoir discriminant relativement à un histogramme couleur classique. Le second, en extérieur et à plus large échelle, témoigne de sa résistance à l'ambiguïté perceptive, et sert d'introduction au chapitre suivant, dans lequel un processus de construction de carte va être présenté.



# Navigation, exploration et représentation de l'environnement

---

## Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>95</b>
<b>5.2</b>	<b>Représentation de l'environnement</b>	<b>96</b>
5.2.1	Les solutions existantes	96
5.2.2	Notre représentation spatiale topologique et photométrique	97
<b>5.3</b>	<b>Processus incrémental de construction de la carte</b>	<b>100</b>
5.3.1	Analyse de la problématique	100
5.3.2	(1) : Le départ d'un noeud topologique	102
5.3.3	(2) : La découverte d'un nouveau lieu	103
5.3.4	(3) et (4) : L'arrivée à un noeud topologique	104
<b>5.4</b>	<b>Localisation dans la carte</b>	<b>110</b>
5.4.1	Localisation topologique du robot	110
5.4.2	Localisation photométrique et gestion des fermetures de boucle	110
5.4.3	Identification de l'orientation des branches du squelette	113
<b>5.5</b>	<b>Déplacements dans l'environnement</b>	<b>113</b>
<b>5.6</b>	<b>Exemple d'application : exploration et couverture de l'espace</b>	<b>115</b>
<b>5.7</b>	<b>Conclusion</b>	<b>117</b>

---

## 5.1 Introduction

Dans les chapitres précédents, nous avons présenté la construction de représentations locales qui permettent à un robot d'appréhender la scène qu'il perçoit. D'un côté, le squelette extrait la structure topologique de l'espace libre proximal, et décrit des trajectoires sûres sur lesquelles il peut évoluer. De l'autre, la signature construite définit une empreinte compacte du lieu observé, et offre donc un outil adapté pour lui permettre de se localiser.

Nous nous intéressons maintenant à la cohérence temporelle et structurelle de ces représentations, et cherchons à les combiner pour offrir au robot des possibilités de comportement autres que réactives. La connaissance a priori étant supposée nulle, nous développons un processus incrémental de construction, qui articule les deux représentations locales sous la forme d'une carte topologique de l'environnement. Fondé exclusivement sur l'information visuelle, cet outil permet au robot non

seulement de se localiser, mais aussi de planifier ses déplacements. Un exemple d'application sera donné en fin de chapitre, sous la forme d'une mission d'exploration et de couverture de l'espace.

## 5.2 Représentation de l'environnement

### 5.2.1 Les solutions existantes

Traditionnellement, les méthodes de cartographie, qu'elles soient fondées sur une information visuelle ou non, peuvent être classifiées suivant deux grandes familles :

- les approches métriques qui définissent une représentation supposée géométriquement exacte de l'environnement (comme par exemple le plan d'une ville à l'échelle, Figure.5.1(a)) ;
- les approches topologiques, qui le modélisent par un ensemble de lieux définis par des noeuds, associé aux relations d'adjacence qui les lient (comme par exemple le plan d'un métro, Figure.5.1(b)).



FIGURE 5.1 – Illustration de la différence entre les approches métriques et topologiques. (a) Exemple métrique : une portion du plan de la ville d'Amiens, (b) exemple topologique : une portion du plan du métro de Paris.

Parmi les approches métriques, de nombreuses solutions sont fondées sur une grille d'occupation. Elles permettent de discrétiser l'environnement sous la forme d'une grille où chaque case est caractérisée par une valeur qui permet de définir si elle représente 1) une portion navigable, 2) une zone impraticable, ou 3) une région encore inconnue. D'abord utilisée à partir de sonars [Elfes 1987, Thrun 2003], cette représentation a depuis été adaptée à de nombreux autres types de capteurs, notamment aux capteurs de vision. Dans [Merveilleux 2013] par exemple, les espaces libres extraits à chaque instant dans une image omnidirectionnelle sont recalés par odométrie et fusionnés dans une telle grille. Dans [Li 2013], les informations fournies par un dispositif de stéréovision sont fusionnées au sein d'une grille multi-résolution. Dans [Nguyen 2012], une approche similaire est couplée à un module de suivi des

objets en mouvement pour produire une grille d'occupation plus réactive, dans un contexte d'assistance à la conduite. D'autres méthodes préfèrent mémoriser dans un référentiel unique la position absolue d'objets d'intérêts identifiés au cours de la mission. Dans le cadre de la vision, il s'agit souvent de points d'intérêt [Biber 2005, Ellekilde 2007].

De leur côté, les approches topologiques cherchent plutôt à représenter l'environnement sous la forme d'un graphe où les noeuds modélisent les différents lieux qui le composent, et les arêtes marquent les relations d'adjacence qui les lient. Il n'existe toutefois aucun consensus sur la signification précise à donner aux noeuds [Kuipers 2004]. L'une des principales difficultés est donc d'en définir le sens physique. Certaines approches [Fabrizi 2000, Tapus 2006, Mozos 2006] détectent les changements marquants dans la topologie locale, comme par exemple le franchissement d'une porte, et définissent un nouveau noeud à chaque fois qu'une telle transition est détectée. La plupart [Cummins 2011, Ulrich 2000, Booij 2006], s'en tient toutefois à la définition proposée par Kuipers [Kuipers 2000], qui considère un noeud comme un ensemble de positions géographiquement proches, présentant une perception identique ou fortement voisine. C'est alors sur la façon dont est définie la ressemblance entre représentations que les travaux diffèrent. Les critères utilisés font écho à l'état de l'art présenté sur les descripteurs visuels dans le chapitre précédent, nous y renvoyons donc le lecteur.

### 5.2.2 Notre représentation spatiale topologique et photométrique

Nous choisissons dans ce travail une approche topologique, et proposons dans un premier temps de construire notre représentation dans l'espace des signatures, en nous basant sur le concept de lieu développé dans le chapitre précédent. Nous introduisons pour cela la notion de **noeud photométrique** :

**Définition 4 (Noeud photométrique)** *Un noeud photométrique  $n^{ph}$ , caractérisé par une signature  $Sig(n^{ph})$ <sup>1</sup>, désigne un ensemble de poses  $p$  de l'environnement dont la perception instantanée  $I_p$  aboutit à une signature  $Sig(I_p)$  fortement similaire à  $Sig(n^{ph})$ , au sens d'un seuil  $\tau_{ph}$  donné. On a :*

$$p \in n^{ph} \iff d(Sig(I_p), Sig(n^{ph})) \geq \tau_{ph} \quad (5.1)$$

L'environnement dans lequel évolue le robot est ainsi divisé en un ensemble  $N^{ph} = \{n_i^{ph}\}$  de noeuds photométriques. Les relations d'adjacence qui les lient sont définies suivant un critère temporel qui traduit les déplacements du robot. Elles sont caractérisées dans la représentation par un élément que nous nommons **tronçon** :

**Définition 5 (Tronçon)** *Un tronçon  $tr$  connecte deux noeuds photométriques  $n_1^{ph}$  et  $n_2^{ph}$ , s'il existe deux instants consécutifs  $t_1$  et  $t_2$  tels que  $p_{t_1} \in n_1^{ph}$  et  $p_{t_2} \in n_2^{ph}$ , ou bien  $p_{t_1} \in n_2^{ph}$  et  $p_{t_2} \in n_1^{ph}$ . On a donc  $d(Sig(n_1^{ph}), Sig(n_2^{ph})) < \tau_{ph}$ .*

1. La signature  $Sig(n^{ph})$  du noeud  $n^{ph}$  est assimilée à celle  $Sig(I)$  construite à partir de l'image  $I$  acquise à l'instant de la création du noeud



L'ensemble  $T = \{tr_i\}$  des tronçons est associé à celui  $N^{ph}$  des noeuds photométriques, pour définir une structure de graphe  $C^{ph} = \{N^{ph}, T\}$  dans l'espace des signatures. Ce type de représentation peut être vue comme une carte topologique (photométrique) de l'environnement, et a été utilisée comme telle au début de cette thèse [Marie 2013b, Marie 2013a].

Une telle structuration de l'information se révèle toutefois insuffisante si l'on souhaite fournir à notre robot une autonomie totale en terme de navigation. Les relations d'adjacence étant définies dans l'espace des signatures, elles ne permettent pas de modéliser les déplacements physiques du robot. Ainsi, tout en sachant qu'il lui est possible de naviguer du noeud  $n_1^{ph}$  sur lequel il se trouve, vers un noeud  $n_2^{ph}$  qui lui est connexe, aucune opération sur  $C^{ph}$  ne permet au robot de transformer cette information en une trajectoire dans l'environnement.

Pour cette raison, nous introduisons un second niveau à notre représentation. L'idée est de proposer une réalisation de  $C^{ph}$  directement dans l'espace navigable, afin d'associer un déplacement concret à chaque tronçon. Nous nous basons pour cela sur le squelette de l'espace libre, qui porte la topologie de l'environnement et peut donc être modélisé par un nouveau graphe  $C^{tp} = \{N^{tp}, A\}$ . Deux éléments s'ajoutent alors à notre représentation : le **noeud topologique**  $n^{tp} \in N^{tp}$ , et l'**arête**  $a \in A$ . Ils sont définis comme suit :

**Définition 6 (Noeud topologique)** *Un noeud topologique  $n^{tp}$  de  $N^{tp}$  désigne l'extrémité d'au moins une voie navigable de l'environnement. Son degré  $d(n^{tp})$ , qui correspond au nombre d'arêtes qui y sont connectées, permet de déterminer<sup>2</sup> s'il marque la présence d'une impasse, caractérisée par  $d(n^{tp}) = 1$ , ou bien d'une intersection, c'est-à-dire  $d(n^{tp}) \geq 3$ .*

**Définition 7 (Arête)** *Une arête  $a$  de  $A$  définit une relation d'adjacence entre deux noeuds topologiques  $n_1^{tp}$  et  $n_2^{tp}$ , et modélise donc une des branches du squelette de l'espace navigable.*

Afin de garantir un lien étroit entre les deux niveaux de représentation  $C^{tp}$  et  $C^{ph}$ , chaque noeud topologique est considéré comme un lieu à part entière. On lui associe par conséquent un noeud photométrique :

$$\forall n_i^{tp} \in N^{tp}, \exists n_j^{ph} \in N^{ph} \mid \text{Sig}(n_i^{tp}) = \text{Sig}(n_j^{ph}) \quad (5.2)$$

Puisque le robot se déplace exclusivement sur le squelette de l'espace libre, tous les autres noeuds photométriques définissent des lieux qui sont situés le long des arêtes  $a$  de  $A$ . Les relations d'adjacence définies dans l'espace des signatures (les tronçons) sont par conséquent assimilables à des portions précises du squelette. Chaque arête  $a$  de l'environnement sert donc de support à un sous-ensemble  $C^a = \{N^a, T^a\}$  de  $C^{ph}$ , où les deux noeuds extrémités  $n_1^a$  et  $n_{|N^a|}^a$  définissent également des noeuds topologiques.

Au final, notre représentation  $C$  est définie comme suit :

---

2.  $d(n^{tp}) = 2$  implique que le robot se trouve sur une arête, et ne peut donc mener à la génération d'un noeud topologique

**Définition 8 (Carte topologique)** Nous définissons la carte  $C$  d'un environnement  $E$  comme un graphe topologique  $\{N^{tp}, A\}$  dont la réalisation est le squelette de l'espace navigable, et où chaque arête  $a$  de  $A$  sert de support à une structure topologique  $\{N^{ph}, T\}$  définie dans l'espace des signatures.

La structure finale de notre représentation  $C$  est résumée Figure 5.3.

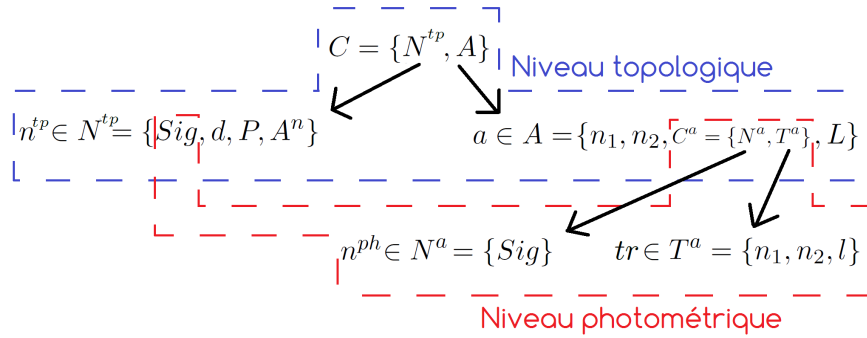


FIGURE 5.2 – Illustration de la carte topologique proposée.

La suite du chapitre sera consacrée à la construction et à la manipulation de cette représentation. Au fil du discours, chacun des attributs portés par les différents éléments sera mis en lumière. Nous expliciterons alors la façon dont ils sont extraits et utilisés. Pour l'heure, nous proposons de nous contenter d'une description rudimentaire. Le niveau topologique  $C^{tp} = \{N^{tp}, A\}$  est défini par une collection  $N^{tp}$  de noeuds, associée à un ensemble  $A$  d'arêtes. Chaque noeud  $n^{tp}$  de  $N^{tp}$  contient les attributs suivants :

Structure d'un noeud topologique $n^{tp}$	
$Sig$	signature photométrique permettant d'identifier le noeud
$d$	degré du noeud (nombre d'arêtes connectées au noeud)
$P$	patch définissant l'orientation de référence du noeud
$A^{n^{tp}} = \{a_i^{n^{tp}}, i = 1, \dots, d\}$	indices dans $A$ des arêtes au départ du noeud

Chaque arête  $a$  de  $A$  porte quant à elle les informations suivantes :

Structure d'une arête $a$	
$L$	longueur estimée de l'arête
$n_1, n_2$	noeuds extrémités de l'arête
$C^a = \{N^a, T^a\}$	structure topologique photométrique portée par l'arête

Le niveau photométrique est composé de noeuds (photométriques) et de tronçons qui portent les informations suivantes :

Structure d'un noeud photométrique $n^{ph}$	
$Sig$	Signature photométrique caractéristique du lieu modélisé

Structure d'un tronçon $tr$	
$l$	longueur estimée du tronçon
$n_1, n_2$	noeuds extrémités du tronçon

## 5.3 Processus incrémental de construction de la carte

### 5.3.1 Analyse de la problématique

Nous nous intéressons maintenant au processus qui aboutit à la construction de la représentation mixte photométrique et topologique proposée. Nous considérons pour cela un environnement  $E$  initialement inconnu, et nous nous plaçons à un instant  $t$  quelconque. Nous notons  $E_t$  la portion de  $E$  qui correspond à l'ensemble des lieux ayant été observés par le robot depuis l'instant  $t_0$ . A défaut de construire la carte  $C$  de  $E$ , qui impliquerait une connaissance totale de l'environnement, l'objectif est pour le robot de disposer d'une représentation  $C_t = \{N_t^{tp}, A_t\}$  qui modélise correctement  $E_t$  (voir Figure 5.3).

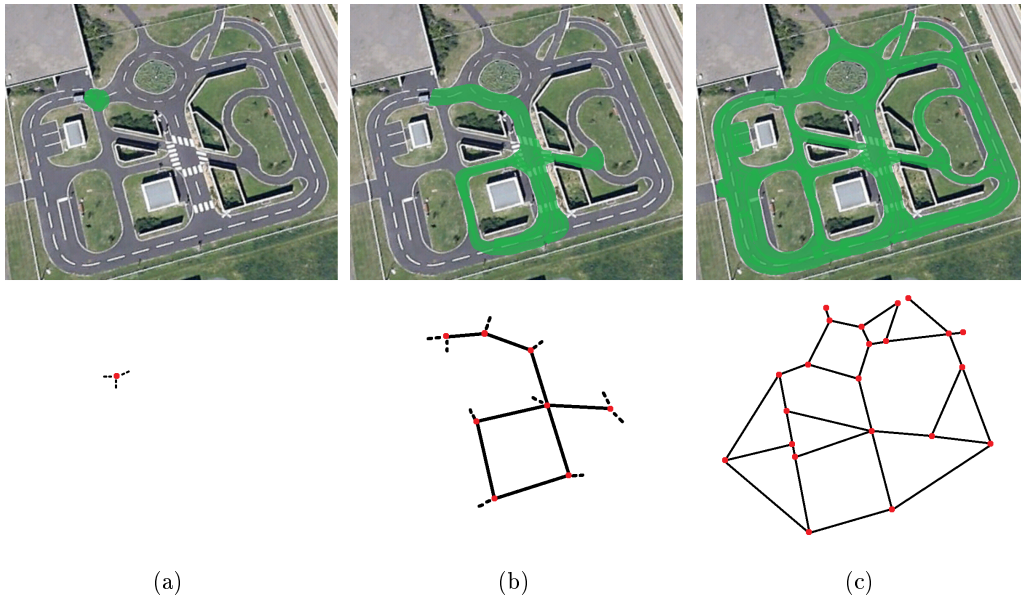


FIGURE 5.3 – Mise en évidence du lien entre la connaissance  $E_t$  du robot sur son environnement (en vert), et la carte  $C_t$  qu'il est capable d'en construire. (a) A l'état initial,  $C_t$  se limite à la perception courante du robot. (b) A un instant  $t$  quelconque,  $C_t$  ne peut modéliser que la portion de l'environnement  $E_t$  connue du robot. (c) Lorsque  $E_t = E$ , la carte  $C_t$  construite n'est autre que la carte  $C$  de  $E$ .

Les observations  $O(I_t)$  du robot construites à partir de l'image acquise  $I_t$ , sont définies par les représentations locales développées dans les chapitres précédents, à savoir le squelette  $Skel(I_t)$  de l'espace libre omnidirectionnel, et la signature photométrique  $Sig(I_t)$  calculée à partir de son complément. Afin d'attribuer une notion

de longueur aux tronçons et arêtes de la représentation, elles sont complétées par les données odométriques  $Odom_t$  déduites dans ce travail des roues codeuses, mais qui pourraient parfaitement être obtenues par un processus d'odométrie visuelle [Scaramuzza 2011b, Scaramuzza 2011a]. On a donc :

$$O(I_t) = [Skel(I_t), Sig(I_t), Odom_t] \quad (5.3)$$

La représentation  $C_t$  est obtenue par la mise en commun des observations  $O(I_0), \dots, O(I_t)$  construites depuis le début de la mission. Pour des raisons évidentes de taille mémoire, il est toutefois inenvisageable de toutes les mémoriser. Tout l'enjeu est donc de mettre en place un processus incrémental capable de construire  $C_{t+1}$  en incorporant  $O(I_t)$  à  $C_t$ . Cette section présente l'élaboration d'un tel processus, avec pour objectif qu'il soit valide en toute circonstance, c'est à dire à tout instant  $t$ , quel que soit l'environnement  $E$ , et peu importe la position que le robot y occupe.

L'environnement du robot est dynamique, mais à ce stade du travail, nous ne considérons pas la mise à jour des connaissances préalablement acquises. Autrement dit, si une modification importante se produit sur une portion de l'environnement déjà modélisée, les noeuds existants ne seront pas modifiés. A la place, de nouveaux seront générés. Cette question de cohérence temporelle globale est l'une des perspectives principales de la thèse, et fera l'objet de nos travaux futurs.

L'organigramme présenté Figure.5.4 illustre les différentes étapes du processus de construction, à savoir :

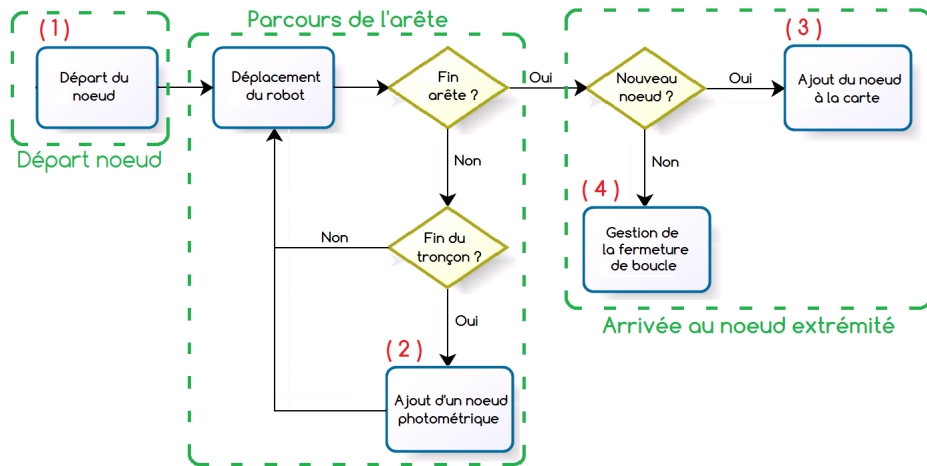


FIGURE 5.4 – Organigramme général du processus d'exploration et de construction d'une arête

- (1) Le départ d'un noeud topologique, où la structure de l'arête est initialisée et connectée au noeud quitté.
- (2) L'arrivée dans un nouveau lieu, qui génère l'ajout d'un noeud photométrique et la création d'un tronçon le connectant au dernier noeud visité.



FIGURE 5.5 – Situation (1) : Départ d'un noeud, le robot s'engage sur une arête inexplorée.

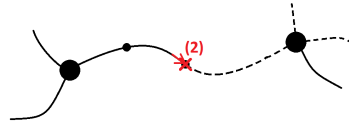


FIGURE 5.6 – Situation (2) : Ajout d'un lieu, le robot se situe sur un nouveau noeud photométrique.

- (3) L'arrivée à un noeud topologique inconnu, qui entraîne sa caractérisation et son ajout à la représentation.

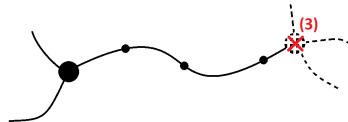


FIGURE 5.7 – Situation (3) : Le robot a atteint un noeud topologique qu'il ne connaît pas.

- (4) L'arrivée à un noeud topologique connu, qui marque une fermeture de boucle et implique une mise à jour du noeud atteint.

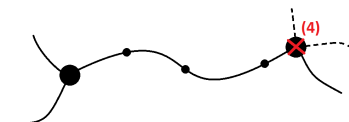


FIGURE 5.8 – Situation (4) : Le robot a atteint un noeud topologique déjà présent dans la carte.

### 5.3.2 (1) : Le départ d'un noeud topologique

Dans cette situation, le robot se trouve sur un noeud topologique  $n^{tp}$ . La procédure débute à l'instant  $t$  où il entame le parcours d'une voie navigable inconnue (voir Fig.5.5). Il s'engage alors sur une branche du squelette qui n'est associée à aucune arête connue. Puisqu'il n'existe dans  $A_t$  aucune structure permettant de mémoriser les observations du robot, l'idée est de générer une arête temporaire, notée  $\hat{a}$ . Elle sera progressivement complétée à mesure que le robot progresse le long de la voie

navigable, et ajoutée à  $A_t$  une fois entièrement définie. Son initialisation est assurée en définissant les éléments connus, à savoir le premier noeud extrémité  $n_1(\hat{a}) = n^{tp}$ . En parallèle, un tronçon temporaire  $\hat{tr}$  est également initialisé ( $n_1(\hat{tr}) = n^{tp}$ ), dans l'attente d'être totalement défini lorsqu'un nouveau lieu (noeud photométrique) sera atteint.

Le noeud  $n^{tp}$  est quant à lui mis à jour afin d'intégrer la relation d'adjacence portée par  $\hat{a}$ . La branche empruntée, associée à un indice  $i$  dans le repère du noeud, y est associée en fixant  $a_i^{n^{tp}} = \hat{a}$ .

Enfin, afin de pouvoir fournir une estimation de longueur aussi bien à  $\hat{tr}$  qu'à  $\hat{a}$ , les données odométriques sont réinitialisées.

La procédure est résumée dans l'algorithme 8.

---

**Algorithme 8** Exploration d'une voie navigable, étape (1) : départ du noeud

---

**Entrées :**

Carte  $C_t = \{N_t^{tp}, A_t\}$  de l'environnement

Noeud courant  $n^{tp}$  de  $N_t^{tp}$

Indice  $i$  de la branche du squelette à explorer

**Sorties :**

Tronçon temporaire  $\hat{tr}$

Arête temporaire  $\hat{a}$

Carte  $C_{t+1} = \{N_{t+1}^{tp}, A_{t+1}\}$  de l'environnement (après mise à jour)

*Définition et initialisation d'un nouveau tronçon :*

$\hat{tr} = \text{new}(\text{Tronçon})$

$n_1(\hat{tr}) = n^{tp}, n_2(\hat{tr}) = \emptyset, l(\hat{tr}) = \emptyset$

*Définition et Initialisation d'une nouvelle arête :*

$\hat{a} = \text{new}(\text{Arête})$

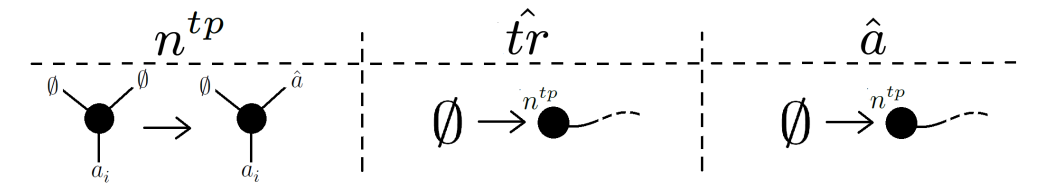
$n_1(\hat{a}) = n^{tp}, n_2(\hat{a}) = \emptyset, L(\hat{a}) = \emptyset$

$N^{\hat{a}} = \{n^{tp}\}, T^{\hat{a}} = \emptyset$

*Mise à jour du noeud  $n^{tp}$  :*

$a_i^{n^{tp}} = \hat{a}$

*Réinitialisation de l'odométrie*



### 5.3.3 (2) : La découverte d'un nouveau lieu

Ce cas de figure correspond à une situation où le robot est en train d'explorer un tronçon  $\hat{tr}$  de l'espace libre. A chaque instant  $t$ , il évalue la similarité

$d(\text{Sig}(I_t), \text{Sig}(n_1(\hat{tr})))$  entre la signature photométrique  $\text{Sig}(I_t)$  calculée à partir de la perception courante, et celle  $\text{Sig}(n_1(\hat{tr}))$  qui caractérise le dernier noeud photométrique créé. Rappelons que cette mesure de similarité, introduite dans le chapitre précédent, fournit une ressemblance définie dans l'intervalle  $[0, 1]$  (1 correspondant à une ressemblance totale). Dans le cas où la valeur obtenue est supérieure à un seuil de similarité  $\tau_{ph}$  fixé, le robot estime ne pas avoir changé de lieu. Dans le cas contraire, il génère un nouveau noeud photométrique  $\hat{n}$  qu'il caractérise à partir de la signature photométrique courante ( $\text{Sig}(\hat{n}) = \text{Sig}(I_t)$ ), avant de l'ajouter à  $N^{\hat{a}}$ . Le tronçon courant  $\hat{tr}$  est alors complètement défini : sa longueur  $l(\hat{tr})$  est obtenue par odométrie, et le noeud extrémité manquant  $n_2(\hat{tr})$  n'est autre que le noeud  $\hat{n}$  qui vient d'être créé. Il peut donc être ajouté à  $T^{\hat{a}}$ . Afin de maintenir la connectivité de la structure topologique  $C^{\hat{a}}$ , un nouveau tronçon temporaire est généré (avec  $\hat{n}$  comme première extrémité), et une réinitialisation de l'odométrie est effectuée.

L'ensemble de la procédure est résumé dans l'algorithme 9.

### 5.3.4 (3) et (4) : L'arrivée à un noeud topologique

Le parcours d'une voie navigable s'achève à l'instant où le robot atteint un noeud topologique. Deux cas de figure sont alors envisageables :

- (3) le robot a atteint un noeud qu'il rencontre pour la première fois, auquel cas il faut l'ajouter à la représentation (voir Fig 5.7) ;
- (4) le robot a atteint un noeud déjà existant et doit par conséquent le mettre à jour (voir Fig 5.8)

#### 5.3.4.1 (3) : L'ajout d'un noeud topologique

Cette situation correspond à un instant  $t$  où le robot atteint un noeud inconnu. Similairement à la situation (2), il commence par générer un noeud  $\hat{n}$  à partir des observations  $O_t$  issues de  $I_t$ . Il s'agit cette fois d'un noeud topologique dont la structure est plus riche que celle d'un noeud photométrique. Les éléments nécessaires à sa définition sont :

- la signature photométrique  $\text{Sig}(\hat{n})$ , qui va servir à correctement localiser le robot lors d'un passage ultérieur au même endroit ;
- le degré  $d(\hat{n})$ , qui établit le nombre d'arêtes connectées au noeud, donc le type de jonction qu'il modélise (impasse, jonction en T, en croix,...) ;
- le patch  $P(\hat{n})$ , qui définit une orientation de référence utilisée dans le processus d'association arête de la représentation / branche du squelette ;
- la liste  $A^{\hat{n}}$  des arêtes connectées  $\hat{n}$ , qui associe à chaque branche perçue, l'arête de  $A_t$  correspondante.

**La signature**  $\text{Sig}(\hat{n})$  est définie par celle calculée à partir de l'image courante. On a  $\text{Sig}(\hat{n}) = \text{Sig}(I_t)$ .

**Le degré**  $d(\hat{n})$  correspond au nombre d'arêtes qui y sont connectées, c'est à dire le nombre de branches du squelette  $\text{Skel}(I_t)$  qui en partent. Pour le déterminer, nous

---

**Algorithme 9** Exploration d'une voie navigable, étape (2) : Arrivée dans un nouveau lieu

---

**Entrées :**

Arête courante  $\hat{a}$   
 Tronçon courant  $\hat{t}r$   
 Signature  $Sig(I_t)$   
 Seuil photométrique  $\tau_{ph}$   
 Distance  $Odom_t$  parcourue depuis la dernière réinitialisation

**Sorties :**

Nouveau tronçon temporaire  $\hat{t}r$   
 Arête temporaire  $\hat{a}$  après mise à jour

*Vérification qu'un nouveau lieu est atteint*

**Si**  $d(Sig(I_t), Sig(n_1(\hat{t}r))) > \tau_{ph}$

Fin

**Sinon**

*Définition et initialisation d'un nouveau noeud photométrique*

$\hat{n} = new(\text{Noeud photométrique})$

$Sig(\hat{n}) = Sig(I_t)$

*Finalisation du tronçon temporaire*

$n_2(\hat{t}r) = \hat{n}, l(\hat{t}r) = Odom_t$

*Mise à jour de l'arête temporaire*

$T^{\hat{a}} = T^{\hat{a}} \cup \hat{t}r, N^{\hat{a}} = N^{\hat{a}} \cup \hat{n}$

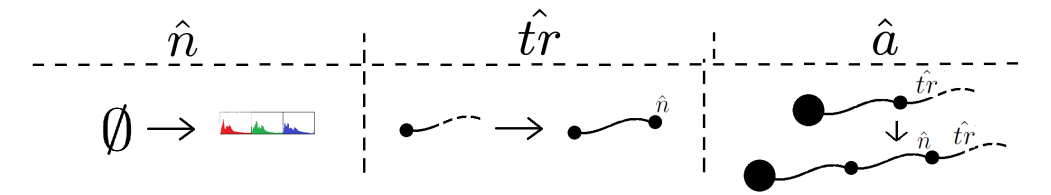
*Définition et initialisation d'un nouveau tronçon*

$\hat{t}r = new(\text{Tronçon})$

$n_1(\hat{t}r) = \hat{n}, n_2(\hat{t}r) = \emptyset, l(\hat{t}r) = \emptyset$

*Réinitialisation de l'odométrie*

**Fin si**





utilisons un ensemble  $\mathcal{B}_t$  (modélisé Figure 5.9), qui est défini dans le plan image normalisé, par l'intersection de la projection du squelette  $Skel(I_t)$  avec le cercle  $\mathcal{C}(c, r)$  de centre  $c = (0, 0)$ , dont le rayon  $r$  permet d'encapsuler la zone aveugle du capteur :

$$\mathcal{B}_t = Skel(I_t) \cap \mathcal{C}(c, r) \quad (5.4)$$

Son cardinal étant égal au nombre de branches du squelette connectées au noeud  $\hat{n}$ , nous fixons  $d(\hat{n}) = |\mathcal{B}_t|$ .

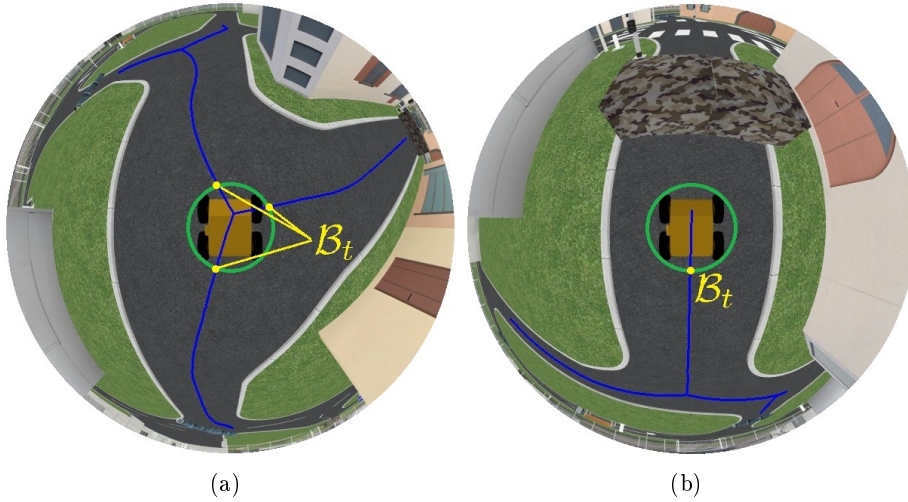


FIGURE 5.9 – Illustration de l'ensemble  $\mathcal{B}_t$  issu de l'intersection du squelette avec le cercle  $\mathcal{C}(c, r)$ , dans le cas (a) d'une intersection entre plusieurs voies navigables, et (b) d'une situation d'impasse.

**Le patch**  $P(\hat{n})$  sert à estimer la rotation relative entre la prise de vue  $I_t$  du noeud à l'instant où il est créé, et l'image acquise lors d'un passage ultérieur du robot. Il s'agit d'un élément crucial, car les voies navigables perçues sont indexées relativement à leur position dans l'image (qui change lorsque le robot effectue une rotation). Une section sera consacrée à cette tâche dans la suite du chapitre. Nous nous contentons pour l'instant d'indiquer que l'information portée par  $P(\hat{n})$  est définie par une bande horizontale dans la configuration  $\mathbf{x}$ . Cette dernière, introduite au chapitre précédent, correspond à la projection de l'image sur le plan  $(\theta, \varphi)$ , tel que présenté Figure 5.10. On a :

$$P(\hat{n}) = \{\mathbf{x}(\theta_P, \varphi), \varphi \in [0, 2\pi]\} \quad (5.5)$$

où  $\theta_P$  est choisie arbitrairement afin de maximiser la pertinence de  $P(\hat{n})$ .

**La liste  $A^{\hat{n}}$  des arêtes connectées au noeud** est le dernier élément à définir. A l'instant où  $\hat{n}$  est créé, seule l'arête  $\hat{a}$  par laquelle le robot l'a atteint est connue. Nous choisissons par convention d'attribuer l'indice 1 à la branche du squelette qui

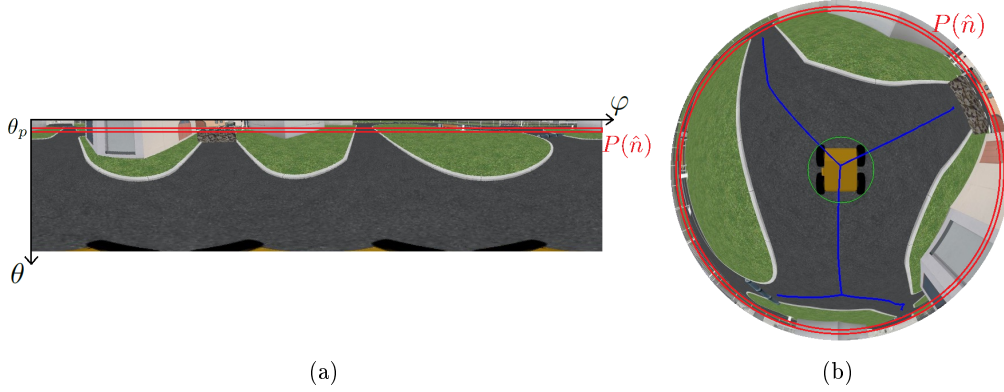


FIGURE 5.10 – Illustration du patch  $P(\hat{n})$  construit pour le noeud  $\hat{n}$  généré. (a) Sur le plan  $(\theta, \varphi)$ , il s'agit d'une bande horizontale définie par l'ensemble des points vérifiant  $\theta = \theta_p$ . (b) Ramenée à l'image omnidirectionnelle, le patch ainsi construit est équivalent à un anneau qui prend donc en considération une information à 360° sur la scène observée.

correspond. Les branches 2 à  $d(\hat{n})$  correspondent quant à elles à des arêtes encore inconnues du robot (faute de quoi le noeud aurait déjà été créé), et sont initialisées à l'ensemble vide. L'ensemble  $A^{\hat{n}}$  des arêtes connectées à  $\hat{n}$  est alors défini par :

$$A^{\hat{n}} = \{a_i^{\hat{n}}, \dots, a_{d(\hat{n})}^{\hat{n}}\}, \text{ avec } a_i^{\hat{n}} = \begin{cases} \hat{a} & \text{si } i = 1 \\ \emptyset & \text{sinon} \end{cases} \quad (5.6)$$

Une fois entièrement défini, le noeud  $\hat{n}$  est ajouté, 1) à la liste  $N^{\hat{a}}$  des noeuds photométriques de  $\hat{a}$ , et 2) à la liste  $N_t^{tp}$  des noeuds topologiques de l'environnement. De leur côté, le tronçon  $\hat{tr}$  et l'arête  $\hat{a}$  sont maintenant délimités ( $n_2(\hat{tr}) = n_2(\hat{a}) = \hat{n}$ ), et peuvent par conséquent être finalisés. La longueur  $l(\hat{tr})$  est obtenue par odométrie, et  $\hat{tr}$  est ajouté à  $T^{\hat{a}}$ . L'arête étant quant à elle formée d'une succession de tronçons,  $L(\hat{a})$  est définie comme la somme de leur longueur. On a :

$$L(\hat{a}) = \sum_{i=1}^{|T^{\hat{a}}|} l(t_i^{\hat{a}}) \quad (5.7)$$

Pour finir, l'arête  $\hat{a}$  est ajoutée à  $A_t$ , marquant ainsi la fin du parcours de la voie navigable. L'ensemble de la procédure est résumé dans l'algorithme 10.

#### 5.3.4.2 (4) : Arrivée à un noeud topologique connu

Le dernier cas correspond à la situation où le robot atteint un noeud topologique qu'il identifie comme étant un noeud  $n^{tp}$  de  $N_t^{tp}$  qu'il a déjà visité. Le tronçon  $\hat{tr}$  et l'arête  $\hat{a}$  sont complétés de manière similaire à ce qui vient d'être présenté pour la situation (3) :  $n^{tp}$  définit les noeuds extrémités  $n_2(\hat{a})$  et  $n_2(\hat{tr})$ , et est ajouté à  $N^{\hat{a}}$  ;

**Algorithme 10** Exploration d'une voie navigable, étape (3) : Arrivée à un noeud topologique inconnu

**Entrées :**

- Carte  $C_t = \{N_t^{tp}, A_t\}$
- Arête courante  $\hat{a}$
- Tronçon courant  $\hat{tr}$
- Signature  $Sig(I_t)$
- Squelette  $Skel(I_t)$
- Configuration  $\mathbf{x}$  (projection de  $I_t$  sur le plan  $(\theta, \varphi)$ )
- Distance  $Odom_t$  parcourue depuis la dernière réinitialisation

**Sorties :**

- Carte  $C_{t+1} = \{N_{t+1}, A_{t+1}\}$  mise à jour

*Définition et initialisation d'un nouveau noeud topologique*

- $\hat{n} = new(\text{Noeud topologique})$
- $Sig(\hat{n}) = Sig(I_t)$
- $d(\hat{n}) = |\mathcal{C}(c, r) \cap Skel(I_t)|$
- $\forall \varphi \in [0, 2\pi], P(\hat{n})[\varphi] = \mathbf{x}(\theta_p, \varphi)$
- $a_1^{\hat{n}} = \hat{a}, a_i^{\hat{n}} = \emptyset, i = 2, \dots, d(\hat{n})$

*Finalisation du tronçon temporaire*

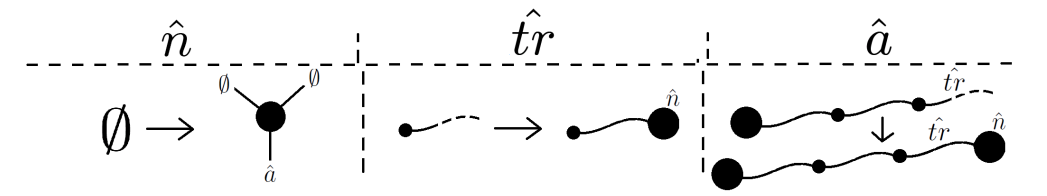
- $n_2(\hat{tr}) = \hat{n}, l(\hat{tr}) = Odom_t$

*Finalisation de l'arête temporaire*

- $n_2(\hat{a}) = \hat{n}$
- $T^{\hat{a}} = T^{\hat{a}} \cup \hat{tr}, N^{\hat{a}} = N^{\hat{a}} \cup \hat{n}$
- $L(\hat{a}) = \sum_{i=1}^{|T^{\hat{a}}|} l(t_i^{\hat{a}})$

*Mise à jour de la carte*

- $A_{t+1} = A_t \cup \hat{a}$
- $N_{t+1} = N_t^{tp} \cup \hat{n}$



la longueur du tronçon  $l(\hat{tr})$  est obtenue par odométrie, tandis que celle de l'arête est obtenue en sommant celle des tronçons portés par  $T^{\hat{a}}$ . Après avoir ajouté  $\hat{tr}$  à  $T^{\hat{a}}$ ,  $\hat{a}$  est ajouté à  $A_t$ .

Le point le plus délicat du processus est toutefois la mise à jour du noeud  $n^{tp}$  identifié. A la différence de l'arrivée à un noeud topologique inconnu, il ne s'agit pas cette fois de définir un nouveau lieu à partir des observations courantes, mais plutôt de modéliser dans  $n^{tp}$  la relation d'adjacence nouvellement établie par le parcours de la voie navigable. Il faut pour cela déterminer l'indice  $i$  de la branche du squelette parcourue, relativement à l'orientation de référence du noeud définie par  $P(n^{tp})$ . Cette procédure est commune avec d'autres tâches liées à la navigation. Elle sera détaillée dans la section consacrée aux modalités de localisation. Pour l'heure, nous supposons l'indice  $i$  connu. La mise à jour de  $n^{tp}$  est réalisée en associant l'arête  $a_i^{\hat{n}}$  de  $A^{\hat{n}}$  à  $\hat{a}$ .

La procédure est résumée dans l'algorithme 11.

---

**Algorithme 11** Exploration d'une voie navigable, étape (4) : Arrivée à un noeud topologique connu

---

**Entrées :**

Carte  $C_t = \{N_t^{tp}, A_t\}$

Arête courante  $\hat{a}$

Tronçon courant  $\hat{tr}$

Signature courante  $Sig(I_t)$

Squelette courant  $Skel(I_t)$

Distance  $Odom_t$  parcourue depuis la dernière réinitialisation

**Sorties :**

Carte  $C_{t+1} = \{N_{t+1}^{tp}, A_{t+1}\}$  mise à jour

*Mise à jour du noeud topologique atteint*

$$a_j^{n_i} = \hat{a}$$

*Finalisation du tronçon temporaire*

$$n_2(\hat{tr}) = n_i, l(\hat{tr}) = Odom_t$$

*Finalisation de l'arête temporaire*

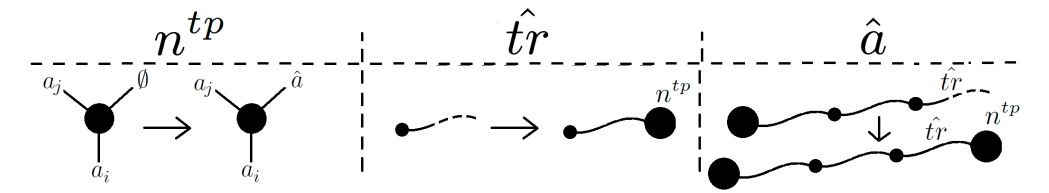
$$n_2(\hat{a}) = n_i$$

$$T^{\hat{a}} = T^{\hat{a}} \cup \hat{tr}, N^{\hat{a}} = N^{\hat{a}} \cup n_i$$

$$L(\hat{a}) = \sum_{i=1}^{l(T^{\hat{a}})} l(t_i^{\hat{a}})$$

*Mise à jour de la carte*

$$A_{t+1} = A_t \cup \hat{a}$$



## 5.4 Localisation dans la carte

Nous introduisons ici l'ensemble des choix algorithmiques qui permettent au robot de se localiser, tant sur un plan photométrique que topologique.

### 5.4.1 Localisation topologique du robot

Que ce soit pour intégrer ses observations courantes à la représentation, ou pour définir la façon dont il se déplace, le robot doit, à chaque instant, déterminer la situation topologique dans laquelle il se trouve. Ainsi, un positionnement sur un noeud topologique déclenchera le processus décisionnel le guidant vers une voie navigable à explorer. A l'inverse, lorsqu'il se situe sur une arête, le robot doit vérifier s'il doit ou non ajouter un lieu à la représentation. Nous proposons de caractériser la situation du robot à un instant  $t$  par un état, noté  $e_t \in \{N, \tilde{N}, A, \bar{S}\}$ , susceptible de prendre l'une des quatre valeurs suivantes :

- $e_t = N$  : le robot se trouve sur un noeud topologique
- $e_t = \tilde{N}$  : le robot est à proximité d'un noeud topologique
- $e_t = A$  : le robot est sur une arête et éloigné de tout noeud topologique
- $e_t = \bar{S}$  : le robot ne se trouve pas sur le squelette de l'espace navigable

Pour caractériser cet état, nous proposons une analyse du squelette  $Skel(I_t)$  fondée sur l'utilisation de deux ensembles  $\mathcal{B}_t$  et  $\mathcal{B}'_t$  définis par son intersection avec deux cercles  $\mathcal{C}(c, r)$  et  $\mathcal{C}(c, r')$  ( $r' > r$ ) centrés sur le robot. Les cardinaux  $|\mathcal{B}_t|$  et  $|\mathcal{B}'_t|$  décrivent le nombre de directions à disposition du robot, donc la topologie de l'espace navigable suivant deux distances différentes. L'état  $e_t$  est alors défini par :

$$e_t = \begin{cases} \bar{S} & \text{si } |\mathcal{B}_t| = 0 \\ A & \text{si } |\mathcal{B}_t| = 2 \ \& \ |\mathcal{B}'_t| = 2 \\ \tilde{N} & \text{si } |\mathcal{B}_t| = 2 \ \& \ |\mathcal{B}'_t| \neq 2 \\ N & \text{sinon} \end{cases} \quad (5.8)$$

La figure 5.11 illustre chacune des quatre situations.

### 5.4.2 Localisation photométrique et gestion des fermetures de boucle

La localisation photométrique est basée sur les signatures de lieu définies dans le chapitre précédent. Elle permet notamment de situer à chaque instant le robot sur l'un des tronçons de l'environnement. La gestion de fermeture de boucle est quant à elle étudiée au niveau des noeuds topologiques. Ces derniers définissent les points de jonction entre les différentes voies navigables, et sont par conséquent susceptibles d'être atteints par le robot à partir d'arêtes différentes.

Considérons la situation où le robot atteint un noeud topologique  $n^{tp}$ . Il s'agit de déterminer s'il rencontre le noeud pour la première fois, et, dans le cas contraire, d'identifier le noeud  $n_i^{tp}$  de  $N_t^{tp}$  qu'il vient d'atteindre. Afin de minimiser les risques

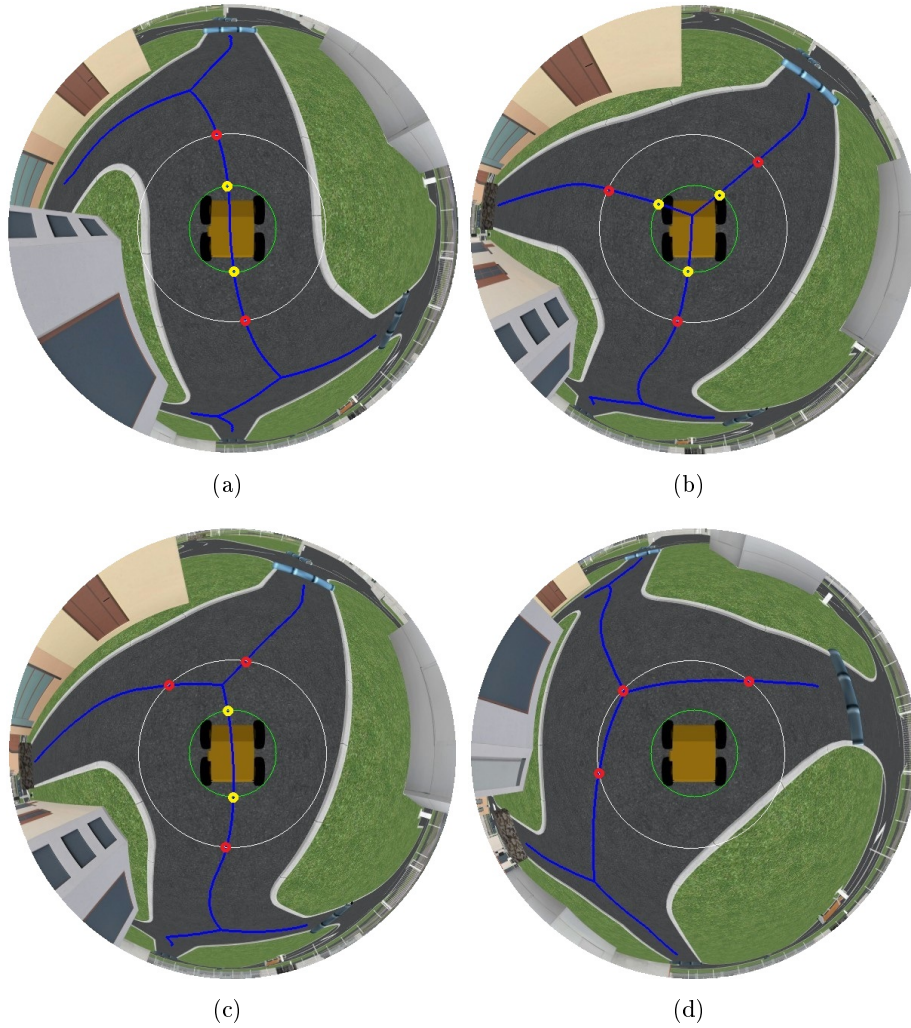


FIGURE 5.11 – Illustration des ensembles  $\mathcal{B}_t$  en **jaune**, et  $\mathcal{B}'_t$  en **rouge**, lorsque le robot est (a) sur une arête, (b) sur un noeud topologique, (c) à proximité d'un noeud topologique, et (d) hors du squelette de l'espace navigable.

d'aliasing perceptif, nous proposons un processus décisionnel hiérarchique fondé sur la combinaison des informations topologiques et photométriques.

Le squelette  $Skel(I_t)$  est dans un premier temps analysé afin de déterminer le nombre de voies navigables connectées à  $n^{tp}$ . Le cardinal  $|\mathcal{B}_t|$  est comparé au degré  $d(n_i^{tp})$  de chaque noeud  $n_i^{tp}$  de  $N_t^{tp}$ , afin de générer la liste  $N_c^{tp}$  des candidats à une éventuelle mise en correspondance :

$$N_c = \left\{ n_i^{tp} \in N_t^{tp} \mid d(n_i^{tp}) = |\mathcal{B}_t| \right\} \quad (5.9)$$

La signature photométrique  $Sig(I_t)$  construite à partir de  $I_t$  est ensuite comparée à celle  $Sig(n_i^{tp})$  caractérisant chaque noeud  $n_i^{tp}$  de  $N_c$ , afin de déterminer le noeud

$n^+$  qui présente la plus forte ressemblance photométrique :

$$n^+ = \arg \max_{n_i^{tp} \in N_c} d(\text{Sig}(I_t), \text{Sig}(n_i^{tp})) \quad (5.10)$$

Deux cas de figure sont alors possibles :

- le noeud  $n^+$  sélectionné n'est pas suffisamment similaire à l'image courante. Le robot estimera alors avoir atteint un nouveau noeud (situation (3)) ;
- la similarité est suffisamment forte pour estimer que  $n^+$  est effectivement le noeud qui vient d'être atteint. Il s'agit dans ce cas d'une situation de fermeture de boucle (situation (4)), avec  $n^{tp} = n^+$ .

Ces deux situations sont différenciées par l'utilisation d'un seuil  $\tau_{tp}$ , qui définit la mesure de similarité minimale permettant d'affirmer que  $n_i^{tp} = n^+$ . On a :

$$n^{tp} = \begin{cases} n^+ & \text{si } d(\text{Sig}(I_t), \text{Sig}(n^+)) \geq \tau_{tp} \\ \hat{n} & \text{sinon} \end{cases} \quad (5.11)$$

La figure 5.12 illustre l'ensemble de la procédure.

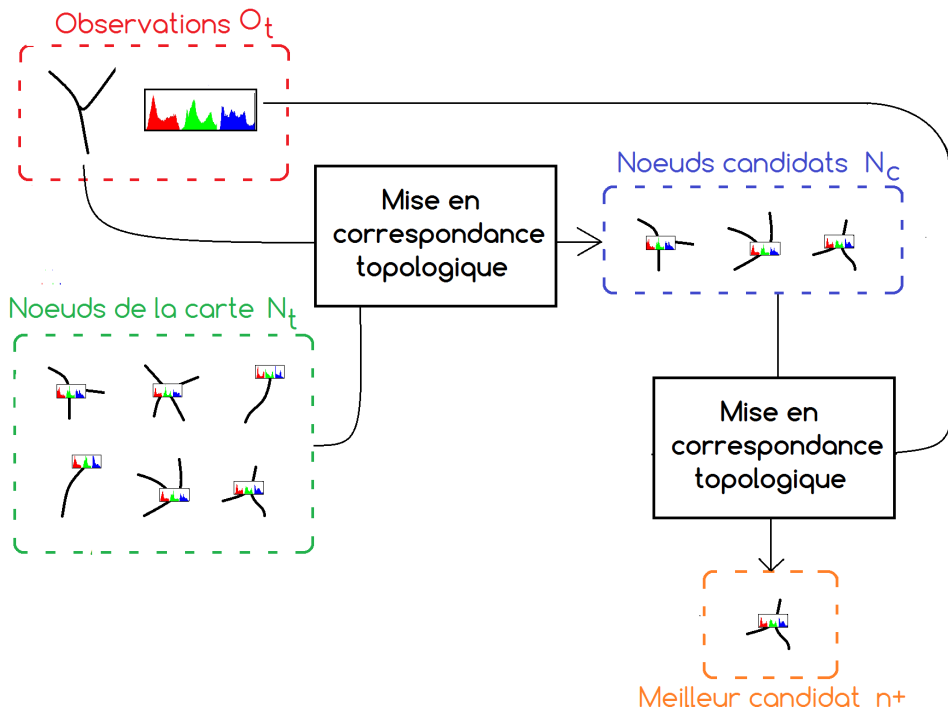


FIGURE 5.12 – Illustration du processus d'identification du noeud atteint.

### 5.4.3 Identification de l'orientation des branches du squelette

Lorsque le robot se trouve sur un noeud topologique  $n^{tp}$  de  $N_t^{tp}$ , il doit pouvoir associer à chaque voie navigable qu'il perçoit et qu'il a déjà visité, l'une des arêtes  $a_i^n$  de  $A^n$  connectées au noeud dans la représentation. Suivant l'orientation du robot, la position des différentes branches du squelette dans l'image subit une rotation autour du centre de projection  $c$ . L'association arête / voie navigable passe donc par l'estimation de l'orientation du robot relativement à une image de référence sur laquelle chaque voie navigable a été identifiée.

Nous proposons pour cela de nous baser sur le compas visuel élaboré dans [Labrosse 2006]. A partir d'une image panoramique de référence  $I_r$ , cet algorithme permet d'évaluer l'orientation relative  $\alpha_r$  d'une image panoramique  $I$  prise approximativement à la même position. Il mesure pour cela la distance euclidienne photométrique qui sépare  $I_r$  et chacune des images  $I_\alpha$  obtenues par application d'une rotation d'angle  $\alpha$  à  $I$ . On a :

$$\alpha_r = \arg \min_{\alpha \in [0, 2\pi]} \sqrt{\sum_i \sum_j (I_r(i, j) - I_\alpha(i, j))^2} \quad (5.12)$$

La vue panoramique est dans notre cas le plan  $(\theta, \varphi)$  utilisé dans le processus de construction de signature. L'algorithme de Labrosse est donc directement applicable. En l'état, il est toutefois trop coûteux aussi bien en temps de calcul qu'en mémoire, puisqu'il suppose de mémoriser pour chaque noeud une vue panoramique complète. Comme suggéré par l'auteur, nous ne tenons compte que d'une portion de la vue panoramique. Nous définissons pour chaque noeud  $n^{tp}$  le patch  $P(n^{tp})$  sous la forme d'une bande horizontale  $\theta = \theta_P$  du plan  $(\theta, \varphi)$ , équivalente à un anneau dans l'image omnidirectionnelle (voir Figure 5.10) :

$$P(n^{tp}) = \{\mathbf{x}(\theta_P, \varphi), \varphi \in [0, 2\pi]\} \quad (5.13)$$

L'angle  $\theta_P$  est choisi judicieusement afin de maximiser la pertinence de l'information portée par le patch, en réduisant notamment la portion d'espace navigable qu'il contient. L'orientation  $\varphi_r$  de la perception instantanée  $I_t$  est alors définie par :

$$\varphi_r = \arg \min_{\varphi_i \in [0, 2\pi]} \| P(n^{tp}) \Big|_{\varphi_i} - P(I_t) \| \quad (5.14)$$

où  $P(I_t)$  désigne le patch extrait de l'image  $I_t$ , et  $P(n^{tp}) \Big|_{\varphi_i}$  correspond au patch  $P(n^{tp})$  après application d'une rotation  $\varphi = \varphi_i$ .

## 5.5 Déplacements dans l'environnement

La représentation proposée est construite autour du postulat que le robot navigue sur le squelette de l'espace navigable. Il peut ainsi associer un déplacement physique dans l'environnement à un déplacement virtuel dans la représentation. A l'image des fonctions tâches développées par Victorino à partir d'observations télémétriques



[Victorino 2003], nous proposons quatre fonctions de navigation qui définissent un comportement pour chacune des situations susceptibles d'être rencontrées par le robot (voir Figure 5.13). Elles sont toutes fondées sur une loi de commande qui permet de converger vers un point objectif défini dans le plan du sol. Ayant présentés les équations de projection inverse plus tôt dans le mémoire, nous ramenons la définition de chacune des fonctions à la sélection, directement dans l'image, d'un point objectif  $p_o$ .

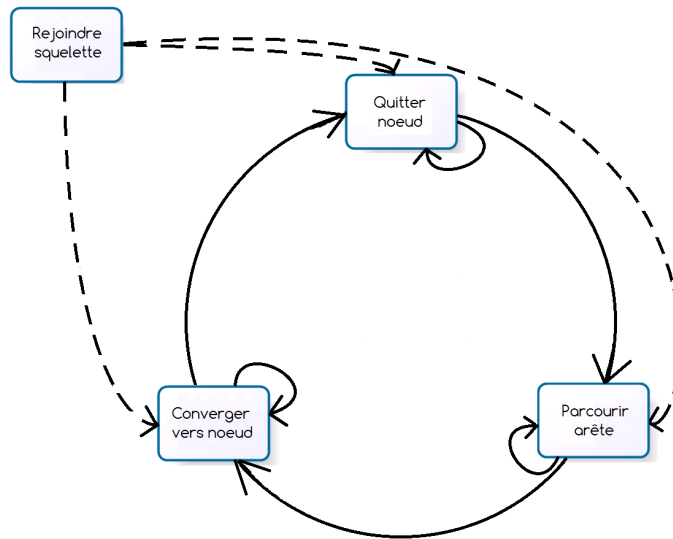


FIGURE 5.13 – Cycle régissant les déplacements du robot. Les quatre tâches présentes regroupent l'ensemble des situations susceptibles d'être rencontrées par le robot.

- **La convergence vers le squelette de l'espace navigable** : Il peut arriver, notamment en début de mission, que le robot ne soit pas positionné sur le squelette de l'espace navigable. Pour le rejoindre, le point  $p_o$  est défini comme le point du squelette  $Skel(I_t)$  le plus proche du centre optique  $c$  de la caméra (donc de la position du robot dans l'image).
- **Le déplacement le long d'une arête** : Dans cette situation, le robot progresse le long d'une branche du squelette, à vitesse régulière, et sans faire demi-tour. Le point objectif  $p_o$  est défini à partir de l'ensemble  $\mathcal{B}_t$  obtenu en intersectant le squelette  $Skel(I_t)$  avec le cercle  $\mathcal{C}(c, r)$ . Les deux points qui composent  $\mathcal{B}_t$  définissent les deux directions que le robot peut emprunter. Le premier, situé dans la partie inférieure de l'image, génèrerait un demi-tour du robot. C'est donc celui situé à l'avant du robot qui est utilisé pour définir  $p_o$ . A noter qu'il s'agit d'un point objectif virtuel, qui se trouve toujours à la même distance du robot tant que ce dernier n'aperçoit pas le noeud topologique vers

lequel il se dirige.

- **La convergence vers un noeud topologique** : Lorsque le robot aperçoit le noeud topologique vers lequel il se dirige, il réduit progressivement sa vitesse afin d'y converger. Pour cela, le point objectif  $p_o$  correspond à la position dans l'image du point de jonction entre les branches du squelette perçues. Il ne s'agit cette fois pas d'un objectif virtuel, mais bien d'un point ancré dans l'environnement, qui s'approche donc du robot tout au long de la phase de convergence.
- **Le départ d'un noeud topologique** : Le départ d'un noeud topologique est un processus qui passe dans un premier temps par l'identification de la branche du squelette que le robot souhaite emprunter. Ce choix dépend de la mission du robot, ainsi que de la stratégie employée pour la mener à terme. L'élément de  $\mathcal{B}_t$  correspondant à la branche sélectionnée est alors utilisée comme point objectif  $p_o$  pour permettre au robot de s'engager sur la voie navigable choisie.

Une illustration du point objectif pour chacune des situations est proposée Figure 5.14.

## 5.6 Exemple d'application : exploration et couverture de l'espace

Pour conclure ce chapitre, nous proposons d'étudier un exemple d'utilisation de notre représentation, dans un contexte de couverture de l'espace. Dans une telle mission, le robot a pour objectif de parcourir au moins une fois chaque voie navigable de l'environnement.

Les directions d'exploration sont définies à partir de la structure topologique qu'il découvre incrémentalement. Chaque fois qu'il atteint un noeud topologique  $n^{tp}$ , il choisit dans la représentation la prochaine arête à parcourir. Ce choix étant guidé par la tâche d'exploration, il doit lui permettre de toujours se diriger vers la branche du squelette inexplorée la plus proche.

Chaque noeud topologique  $n^{tp}$  contient un ensemble  $A^{n^{tp}} = \{a_1^{n^{tp}}, \dots, a_{d(n^{tp})}^{n^{tp}}\}$  définissant les arêtes qui y sont connectées. Lorsqu'une branche du squelette au départ du noeud n'a pas encore été explorée, au moins un élément  $a_i^{n^{tp}}$  de  $A^{n^{tp}}$  est égal à l'ensemble vide. Afin d'identifier la voie navigable inexplorée la plus proche, le robot commence donc par générer la liste  $N_d^{tp}$  des noeuds de  $N_t^{tp}$  qui ont une arête inexplorée :

$$N_d^{tp} = \{n_i^{tp} \in N_t^{tp} \mid \exists j \in [1, \dots, d(n_i^{tp})], a_j^{n_i^{tp}} = \emptyset\} \quad (5.15)$$

Deux cas de figure sont alors envisageables :

- $n^{tp} \in N_d^{tp}$  : le robot se trouve sur un noeud duquel part une voie navigable inconnue. Il s'oriente dans ce cas vers la branche du squelette qui correspond. Si plusieurs branches sont inexplorées, il choisit à ce stade des développements

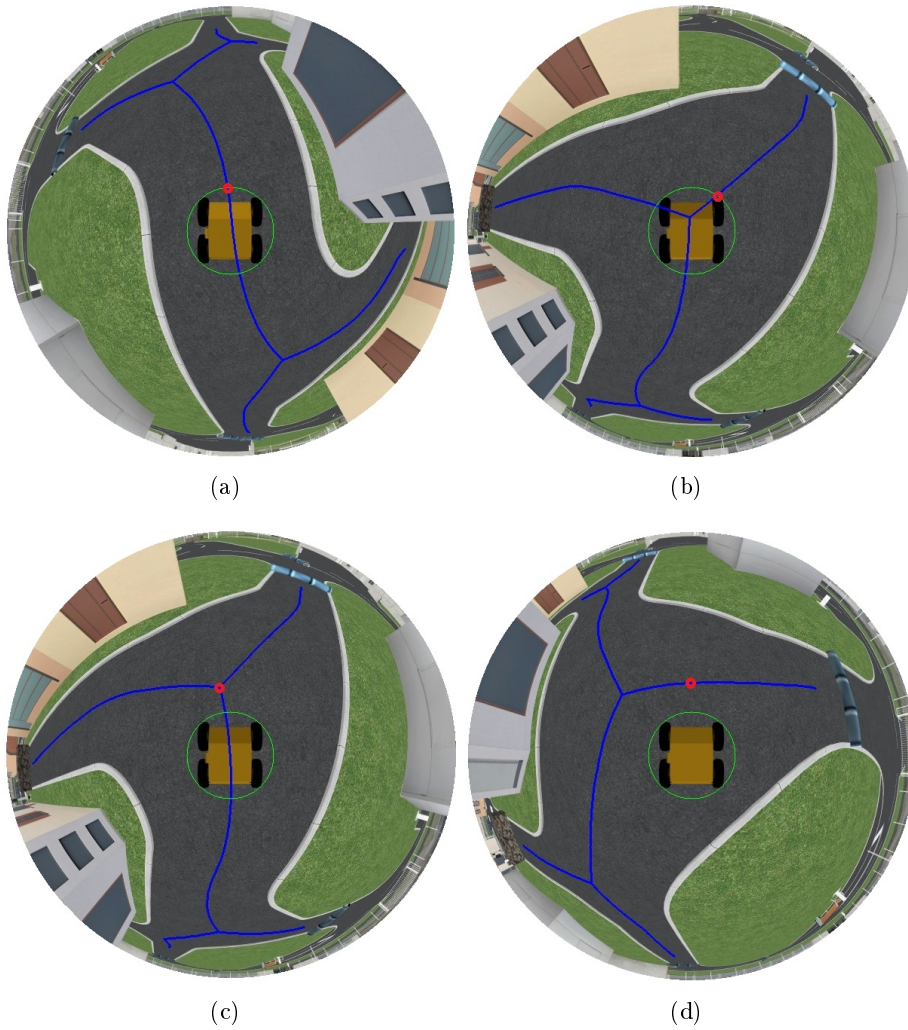


FIGURE 5.14 – Mise en évidence (en rouge), du point objectif  $p_o$  choisi, lorsque le robot est (a) sur une arête, (b) sur un noeud topologique, (c) à proximité d'un noeud topologique, et (d) hors du squelette de l'espace navigable.

la plus proche au sens de l'orientation du robot (d'autres critères sont bien entendu envisageables).

- $n^{tp} \notin N_d^{tp}$  : le robot se trouve sur un noeud où chaque voie navigable connectée est connue. Il doit dans ce cas s'orienter vers la branche du squelette qui le conduit au noeud  $n_d^{tp}$  de  $N_d^{tp}$  le plus proche.

Dans la seconde situation, le noeud  $n_d^{tp}$  le plus proche est défini à partir de l'évaluation du plus court chemin (déterminé par exemple par l'algorithme de Dijkstra) connectant  $n^{tp}$  et l'un des noeuds de  $N_d^{tp}$  :

$$n_d^{tp} = \arg \min_{n_i^{tp} \in N_d^{tp}} d(n^{tp}, n_i^{tp}) \quad (5.16)$$

Le plus court chemin, noté  $It^*(n^{tp}, n_d^{tp})$ , est formé d'une succession d'arêtes  $a_{k_i}$  de  $A_t$ . La première,  $a_{k_1}$ , est connectée à  $n^{tp}$ , donc présente dans  $A^{n^{tp}}$ . Le robot s'engage alors sur la branche correspondante.

Pour illustrer l'ensemble de la procédure, nous proposons d'utiliser le site expérimental Pavin<sup>3</sup>. La portion de l'environnement considérée, son graphe topologique, ainsi qu'une illustration du processus complet d'exploration sont présentés Figure 5.15.

Le graphe topologique est extrait automatiquement par l'algorithme présenté chapitre 3. Nous cherchons à confronter ce résultat à la représentation photométrique et topologique de l'environnement que le robot est en mesure de générer en appliquant la stratégie de couverture de l'espace qui vient d'être présentée. Nous utilisons pour cela le logiciel Cobaye, un outil de prototypage d'applications robotique<sup>4</sup> développé par la société 4D-Virtualiz. Il permet d'appliquer l'ensemble de nos algorithmes sur un robot virtuel placé dans le modèle 3D du site de Pavin. Ce logiciel est encore en phase de développement, et la dynamique du robot y est encore imprécise. C'est cette raison, et non la mise en défaut de nos algorithmes, qui fait que seule une portion du site expérimental est utilisée ici. La figure 5.16 présente la position des différents noeuds (photométriques et topologiques) générés par le robot lors de sa mission, ainsi que les relations d'adjacence qui les lient, pour des seuils  $\tau_{ph}$  et  $\tau_{tp}$  fixés à 0.95.

La mission d'exploration est réalisée de manière totalement autonome par le robot, et la représentation finale obtenue est cohérente avec l'environnement. On en conclut que les différents traitements (construction incrémentale de la carte, processus de localisation, gestion des déplacements et choix des arêtes parcourues) ont eu les résultats attendus.

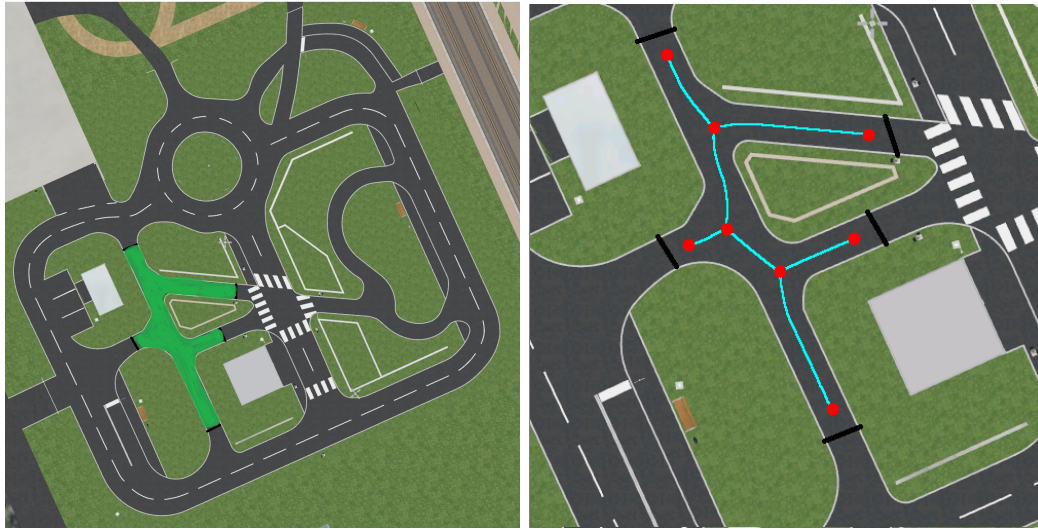
De plus, en comparant ce résultat avec la figure 5.15(b), on remarque qu'il existe un lien fort avec la carte construite depuis une vue aérienne de l'environnement. Les mêmes noeuds topologiques sont créés, et les relations d'adjacence qui les lient sont identiques. Ce constat laisse entrevoir des perspectives très intéressantes, car il sous-entend la possibilité de mettre en place un processus de mise en correspondance entre une vue aérienne d'un environnement, et la perception au sol d'un robot qui s'y déplace.

## 5.7 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'organisation temporelle et structurelle des représentations locales développées depuis le début du mémoire.

3. Pavin est un site expérimental grandeur réelle de robotique situé à Clermont-Ferrand

4. Nous en profitons pour remercier le département ISPR de l'institut Pascal, et la société 4D-Virtualiz pour nous avoir permis d'utiliser cet outil.



(a)

(b)

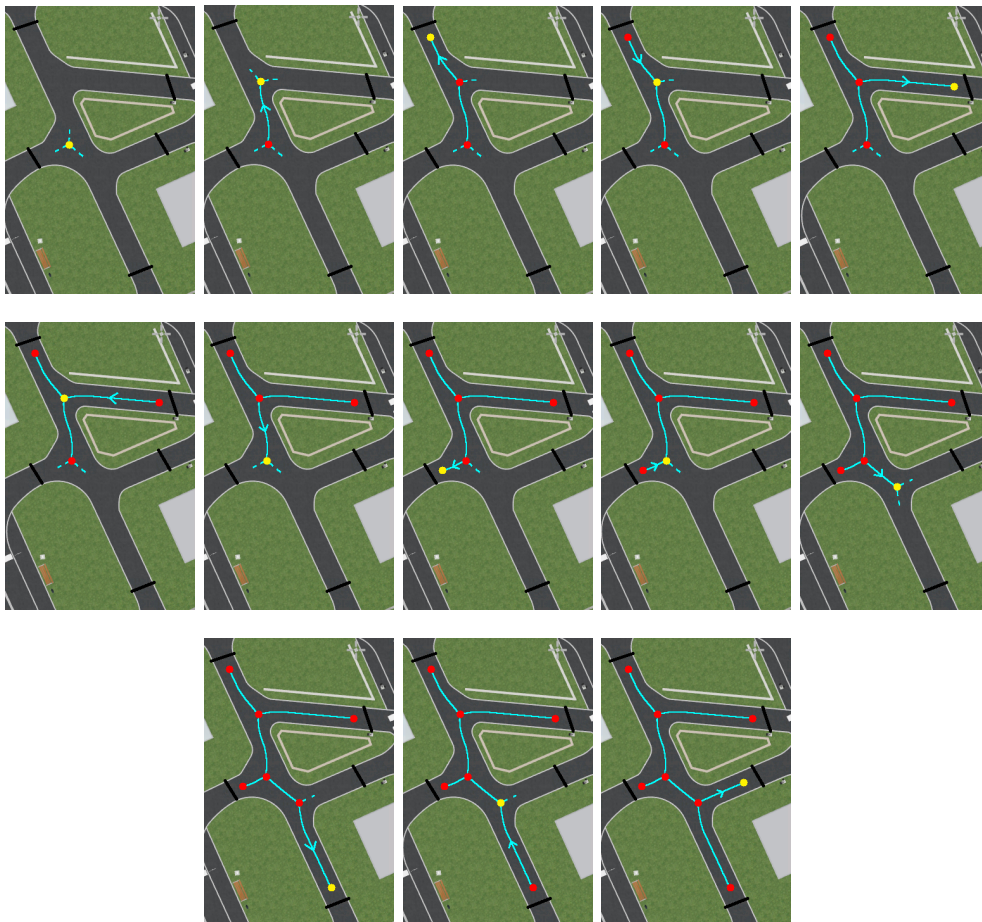


FIGURE 5.15 – (a) Vue aérienne du site expérimental de Pavin, et portion de l'environnement considérée (en vert). (b) Topologie de l'environnement calculée automatiquement à partir de l'algorithme présenté chapitre 3. De haut en bas et de gauche à droite : Illustration de la stratégie d'exploration adoptée par le robot. Le noeud de couleur jaune correspond à chaque fois à la position courante du robot.



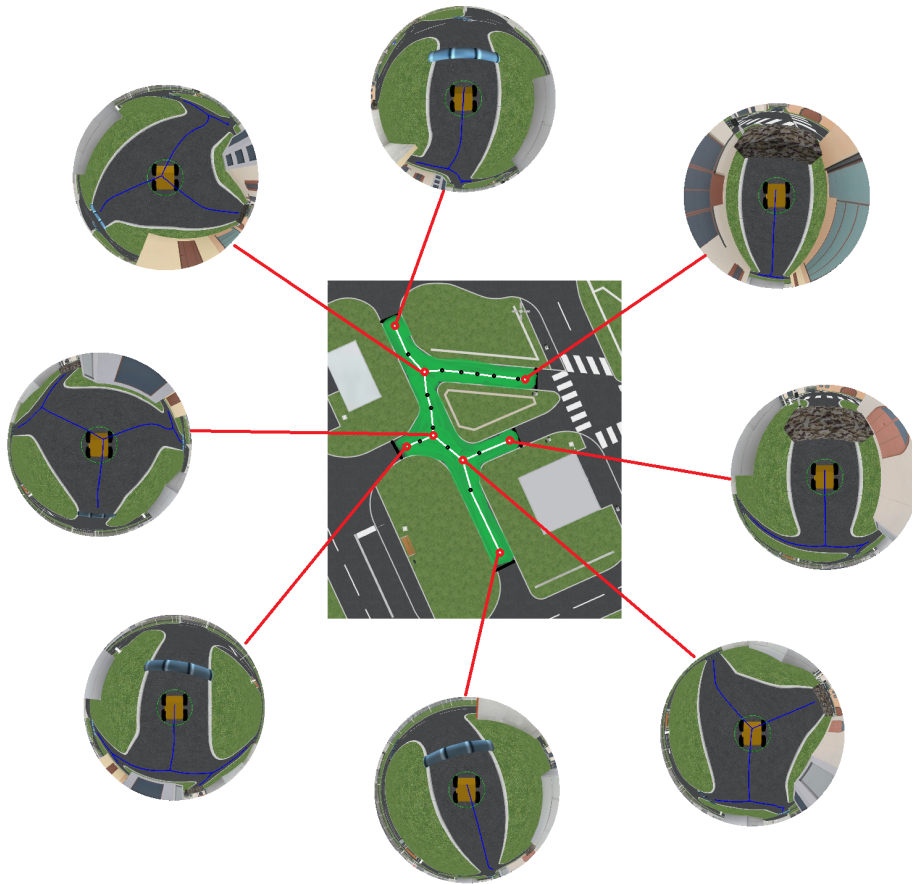


FIGURE 5.16 – Représentation topologique construite automatiquement par le robot en réalisant la mission de couverture de l'espace dans le modèle 3D du site expérimental Pavin (logiciel Cobaye). Les noeuds topologiques (en **rouge**) comme photométriques (en **noir**), ont été positionné manuellement sur une vue aérienne du scénario, en accord avec la position du robot à l'instant où il les crée. Les tronçons de la représentation sont modélisés en **blanc**. Les imagerie correspondent à la perception du robot sur chacun des noeuds topologiques générés.

Afin de fournir au robot un outil sur lequel fonder des processus décisionnels de haut niveau, nous avons élaboré une représentation topologique de l'environnement couplée à une stratégie lui permettant de la construire incrémentalement au fil de ses déplacements. Guidé par le squelette de l'espace navigable (extrait à chaque instant en utilisant le *Delta Medial Axis*), le robot évolue dans une structure de graphe qui sert de support à la représentation. Les lieux qu'il rencontre, caractérisés par la signature proposée, sont indexés directement dans cette structure et permettent au robot de se localiser dans l'environnement à partir de sa perception instantanée. Un exemple d'application est proposé, sous la forme d'une mission de couverture de

l'espace, et une première expérimentation en environnement de synthèse est réalisée pour l'illustrer. Les résultats obtenus sont encourageants, et mettent en évidence la validité de l'approche. Des tests en situation réelle seront par conséquent réalisés dès que les conditions météo le permettront.

# Conclusion générale et perspectives

## Sommaire

<b>6.1</b>	<b>Bilan général</b>	<b>121</b>
<b>6.2</b>	<b>Perspectives</b>	<b>122</b>
<b>6.3</b>	<b>Valorisation des travaux de thèse</b>	<b>123</b>

## 6.1 Bilan général

Dans ce mémoire de thèse, nous avons abordé les questions de la navigation et de la cartographie autonomes de l'environnement, pour un robot mobile terrestre muni d'un capteur de vision omnidirectionnelle. L'image acquise à chaque instant permet de générer différentes représentations locales, qui offrent au robot une compréhension de la scène dans laquelle il évolue, tant sur le plan photométrique que topologique. Afin d'offrir des possibilités comportementales de haut niveau (comme par exemple l'exploration autonome), ces représentations sont ensuite structurées sous la forme d'une carte topologique chargée de modéliser l'environnement.

Dans un premier chapitre, nous nous sommes d'abord attachés à présenter et modéliser le capteur catadioptrique utilisé, afin de bien comprendre les lois qui régissent la projection de la scène dans l'image. Nous avons ensuite introduit un algorithme qui sert de base à la plupart des contributions de cette thèse. En procédant à l'extraction de l'espace navigable, ce dernier permet en effet de segmenter l'image en deux régions aux propriétés très différentes : le sol, d'un côté, qui à travers sa forme porte les propriétés topologiques de l'espace libre ; et l'espace non navigable (les obstacles) de l'autre, qui quant à lui porte la majorité des informations photométriques pertinentes de la scène perçue.

Les deux chapitres suivants se sont intéressés chacun à l'une de ces régions, en proposant une structuration de l'information sous la forme de représentations locales. D'abord, nous nous sommes concentrés sur l'espace navigable, dont la topologie a été caractérisée à partir de son squelette. Dans le souci de prendre en compte les déformations liées au processus d'extraction, nous avons développé notre propre algorithme de squelettisation : le *Delta Medial Axis*. A partir d'un seul paramètre, il permet de calculer une version filtrée du squelette d'une forme discrète 2D, avec



des performances computationnelles en adéquation avec la contrainte temps réel de déplacement du robot. A travers une étude comparative avec des méthodes récentes de l'état de l'art, nous avons pu montrer la plus-value de cet outil, y compris dans des cadres applicatifs plus variés. Afin de définir une topologie en adéquation avec la scène perçue, nous avons dans un second temps intégré, directement dans le processus de squelettisation, les déformations induites par le capteur omnidirectionnel. De cette façon, l'algorithme calcule la projection du squelette de l'espace navigable dans l'image, et non le squelette de sa projection. Par une analyse comparative entre les deux méthodes, nous avons pu mettre en évidence à quel point la différence est importante, en soulignant notamment l'apport de la version adaptée en termes de sécurisation des déplacements

Le chapitre suivant s'est focalisé sur le complément de l'espace navigable, en s'en servant comme support pour caractériser photométriquement la scène perçue. Nous avons ainsi élaboré une signature visuelle fondée sur la théorie des invariants de Haar, qui tient compte de la géométrie particulière du capteur pour générer une empreinte de lieu à la fois invariante et discriminante. En intérieur comme en extérieur, nous avons vu qu'elle pouvait être utilisée comme index pour localiser le robot, et aboutir à un modèle topologique de l'environnement directement dans l'espace des signatures.

Le dernier chapitre a été consacré à la structuration temporelle et spatiale des représentations locales. Nous avons montré qu'il était possible d'indexer chaque lieu (au sens des signatures), dans une structure de graphe portée par le squelette de l'espace navigable. De cette façon, les relations d'adjacence correspondent à des portions précises du squelette, sur lequel le robot est capable de générer des trajectoires d'exploration. Les premiers résultats obtenus en environnement de simulation sont convaincants, la carte topologique construite étant cohérente d'une part avec les déplacements réalisés par le robot, et d'autre part avec celle obtenue à partir d'une vue aérienne de la scène.

## 6.2 Perspectives

Les perspectives que laisse entrevoir ce travail sont nombreuses :

- Pour l'instant, les représentations locales sont traitées de manière complètement indépendantes avant d'être mises en commun lors du processus de construction de carte. Nous pensons que leur qualité respective serait améliorée si les deux étaient considérées conjointement. D'un côté, le squelette pourrait être utilisé pour extraire dans l'image une version filtrée de l'espace navigable (en utilisant la transformée en axe médian). De cette façon, un éventuel obstacle mal détecté n'aurait que peu, voire pas d'impact sur la signature construite. De l'autre, la localisation du robot fournie par la signature pourrait être associée à une topologie attendue, de telle sorte que le squelette extrait ne soit pas complètement erroné suite à une mauvaise segmentation de l'espace navigable.

- Dans un même souci de rendre plus robuste le résultat final, il serait également intéressant de raffermir la base des traitements, en garantissant un espace libre extrait à la fois stable et précis. Pour y parvenir, on pourrait par exemple envisager de coupler le capteur omnidirectionnel avec d'autres dispositifs de perception. Sans changer la philosophie de l'approche, il serait également probablement productif de remplacer le détecteur de contours utilisé (Canny-Deriche), par une approche prenant en compte les spécificités du contexte
- Une autre perspective de ce travail est la mise à jour de la carte sur un horizon temporel important. A ce stade des développements, le robot n'est en effet pas en mesure de modéliser les modifications survenues dans l'environnement entre deux passages dans un même lieu. Le caractère dynamique de l'environnement nécessite donc d'adapter à la fois la représentation et sa manipulation pour permettre au robot de correctement le modéliser.
- Un autre axe de recherche majeur est l'extension de ce travail à un contexte multi-robots ; en particulier en termes de partage et de fusion des représentations. Ces dernières ayant été pensées dans cet esprit, il s'agit d'une extension naturelle des travaux de thèse, qui sera d'ailleurs très rapidement abordée.
- Enfin, dans un registre un peu différent, nous pouvons remarquer que la représentation construite par le robot à partir uniquement de sa perception locale, porte en fin de mission la structure topologique de l'environnement. On peut donc envisager un processus de mise en correspondance avec la carte du même environnement extraite par exemple depuis une vue aérienne.

### 6.3 Valorisation des travaux de thèse

Les travaux présentés dans ce mémoire ont été valorisés par l'intermédiaire de plusieurs vecteurs. Il y a d'abord les publications, nationales comme internationales qui ont jalonné ces trois années. En voici la liste exhaustive :

- Romain Marie, Ouiddad Labbani-Igbida et El Mustapha Mouaddib, *Invariant signatures for omnidirectional visual place recognition and robot localization in unknown environments*, IEEE International Conference on Pattern Recognition, ICPR 2012,
- Romain Marie, Ouiddad Labbani-Igbida, Pauline Merveilleux et El Mustapha Mouaddib, *Autonomous robot exploration and cognitive map building in unknown environments using omnidirectional visual information only*, IEEE Workshop On Robotic Vision, WORV 2013,
- Romain Marie, Ouiddad Labbani-Igbida et El Mustapha Mouaddib, *The Delta-Medial Axis : A robust and linear time algorithm for Euclidian skeleton computation*, IEEE International Conference on Image Processing, ICIP 2013,

- Romain Marie, Ouiddad Labbani-Igbida et El Mustapha Mouaddib, *Exploration autonome et construction de carte cognitive en environnement inconnu basées vision omnidirectionnelle uniquement*, Congrès des jeunes chercheurs en vision par ordinateur, ORASIS 2013,
- Romain Marie, Ouiddad Labbani-Igbida et El Mustapha Mouaddib, *Scale space and free space topology analysis for omnidirectional images*, IEEE International Conference on Robotics and Automation, ICRA 2014,
- Romain Marie, Ouiddad Labbani-Igbida et El Mustapha Mouaddib, *Exploration autonome et construction de carte cognitive en environnement inconnu basées vision omnidirectionnelle uniquement*, numéro spécial TS, 2014

Il y a ensuite les communications orales et séjours scientifiques. En voici quelques exemples :

- Intégration logicielle, site expérimentale de Pavin (Clermont-Ferrand), dans le cadre du projet R-Discover, 20 et 21 Juin 2011.
- Séminaire de recherche, laboratoire XLim, équipe mécatronique, Limoges, 6 Avril 2012.
- Communication orale à la première rencontre UFECAP PRES MIS/CReSTIC, Laon, 20 Décembre 2012.
- Journée d'étude et d'échanges des chercheurs en Image, Vision et Reconnaissance des formes de la Région Nord Pas-de Calais et Picardie, présentation orale, Valenciennes, 6 Juin 2013.
- Organisation et participation (orale et posters), aux Journées des Jeunes Chercheurs du MIS (JJCMIS), Amiens, éditions 2011, 2012 et 2013.

# Bibliographie

- [Ahuja 1997] Narendra Ahuja et Jen-Hui Chuang. *Shape Representation Using a Generalized Potential Field Model*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 2, pages 169–176, 1997. (Cité en page 27.)
- [Andreasson 2005] Henrik Andreasson, André Treptow et Tom Duckett. *Localization for Mobile Robots using Panoramic Vision, Local Features and Particle Filter*. In ICRA, pages 3348–3353, 2005. (Cité en page 63.)
- [Arcelli 1993] Carlo Arcelli et Gabriella Sanniti di Baja. *Euclidean skeleton via centre-of-maximal-disc extraction*. Image Vision Comput., vol. 11, no. 3, pages 163–173, 1993. (Cité en page 27.)
- [Attali 2001] D. Attali et J. o. Lachaud. *Delaunay Conforming Iso-surface, Skeleton Extraction and Noise Removal*, 2001. (Cité en page 29.)
- [Bai 2007] Xiang Bai, Longin Jan Latecki et Wenyu Liu. *Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 3, pages 449–462, 2007. (Cité en page 27.)
- [Baker 1999] Simon Baker et Shree K. Nayar. *A Theory of Single-Viewpoint Cata-dioptric Image Formation*. International Journal of Computer Vision, vol. 35, no. 2, pages 175–196, 1999. (Cité en page 11.)
- [Barla 2003] Annalisa Barla, Francesca Odone et Alessandro Verri. *Histogram intersection kernel for image classification*. In ICIP (3), pages 513–516, 2003. (Cité en page 83.)
- [Barreto 2001] J. P. Barreto et H. Araújo. *Issues on the geometry of central cata-dioptric image formation*. In CVPR, pages 422–427, 2001. (Cité en pages 8, 11, 13 et 67.)
- [Bertrand 1995] Gilles Bertrand. *A parallel thinning algorithm for medial surfaces*. Pattern Recognition Letters, vol. 16, no. 9, pages 979–986, 1995. (Cité en page 26.)
- [Bertrand 2008] Gilles Bertrand et Michel Couprie. *Two-Dimensional Parallel Thinning Algorithms Based on Critical Kernels*. Journal of Mathematical Imaging and Vision, vol. 31, no. 1, pages 35–56, 2008. (Cité en page 27.)
- [Biber 2005] Peter Biber et Tom Duckett. *Dynamic Maps for Long-Term Operation of Mobile Service Robots*. In Robotics : Science and Systems, 2005. (Cité en page 97.)
- [Blum 1964] H. Blum. *A transformation for extracting new descriptors of shape*. In WWD, editeur, Models for the Perception of Speech and Visual Form, pages 362–380. MIT Press, 1964. (Cité en page 22.)
- [Blum 1978] Harry Blum et Roger N. Nagel. *Shape description using weighted symmetric axis features*. Pattern Recognition, vol. 10, no. 3, pages 167–180, 1978. (Cité en page 29.)

- [Booij 2006] Olaf Booij, Zoran Zivkovic et Ben J. A. Kröse. *Sparse appearance based modeling for robot localization*. In IROS, pages 1510–1515, 2006. (Cité en page 97.)
- [Booij 2007] Olaf Booij, Bas Terwijn, Zoran Zivkovic et Ben J. A. Kröse. *Navigation using an appearance based topological map*. In ICRA, pages 3927–3932, 2007. (Cité en page 63.)
- [Borgefors 1989] G Borgefors, I Ragnemalm et G. Sanniti di Baja. *A medical axis transform algorithm for compression and vectorization of document images*. In Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, pages 1850–1853 vol.3, 1989. (Cité en page 27.)
- [Brandt 1991] J.W. Brandt et A. Jain. *The Euclidian Distance Transform : finding the local maxima and reconstructing the shape*. In 7th Scandinavian Conference on Image Analysis, pages 974–981 vol.2, 1991. (Cité en page 29.)
- [Canny 1986] John Canny. *A computational approach to edge detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, no. 6, pages 679–698, 1986. (Cité en page 16.)
- [Caron 2011] Guillaume Caron et Damien Eynard. *Multiple camera types simultaneous stereo calibration*. In ICRA, pages 2933–2938, 2011. (Cité en page 14.)
- [Charron 2006] Cyril Charron, Ouidad Labbani-Igbida et El Mustapha Mouaddib. *Constitution active de percepts spatiaux, application à la reconnaissance de lieux*. In Thèse, Université de Picardie Jules Verne, 2006. (Cité en pages 65, 66, 68, 69 et 81.)
- [Chaussard 2011] J. Chaussard, M. Couprie et H. Talbot. *Robust skeletonization using the discrete  $\lambda$ -medial axis*. Pattern Recognition Letters, vol. 32, no. 9, 2011. (Cité en pages 25, 27, 28, 29, 32, 33, 38, 40, 41 et 43.)
- [Chazal 2005] Frédéric Chazal et André Lieutier. *The "Lambda-Medial Axis"*. Graph. Models, vol. 67, no. 4, pages 304–331, Juillet 2005. (Cité en pages 25 et 29.)
- [Chuang 2000] Jen-Hui Chuang, Chi-Hao Tsai et Min-Chi Ko. *Skeletonization of Three-Dimensional Object Using Generalized Potential Field*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 11, pages 1241–1251, 2000. (Cité en page 27.)
- [Coeurjolly 2007] David Coeurjolly et Annick Montanvert. *Optimal Separable Algorithms to Compute the Reverse Euclidean Distance Transformation and Discrete Medial Axis in Arbitrary Dimension*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 3, pages 437–448, 2007. (Cité en pages 29 et 34.)
- [Cohen 1991] Laurent D. Cohen. *On active contour models and balloons*. CVGIP : Image Underst., vol. 53, no. 2, pages 211–218, Mars 1991. (Cité en page 16.)
- [Couprie 2007] M. Couprie, D. Coeurjolly et R. Zrour. *Discrete bisector function and Euclidean skeleton in 2D and 3D*. Image and Vision Comp., vol. 25, no. 10, pages 1519–1698, 2007. (Cité en pages 26, 27, 29, 32 et 33.)

- [Cummins 2011] Mark Cummins et Paul M. Newman. *Appearance-only SLAM at large scale with FAB-MAP 2.0*. International Journal of Robotic Research, pages 1100–1123, 2011. (Cité en page 97.)
- [Danielsson 1980] P. Danielsson. *Euclidean distance mapping*. Computer Graphics and Image Processing, vol. 14, no. 3, pages 227–248, 1980. (Cité en page 51.)
- [Davies 1981] Edward Roy Davies et A. P. N. Plummer. *Thinning algorithms : A critique and a new methodology*. Pattern Recognition, vol. 14, no. 1-6, pages 53–63, 1981. (Cité en page 28.)
- [Deriche 1987] Rachid Deriche. *Using Canny's criteria to derive a recursively implemented optimal edge detector*. International journal of computer vision, vol. 1, no. 2, pages 167–187, 1987. (Cité en page 16.)
- [Deselaers 2008] Thomas Deselaers. Image retrieval, object recognition, and discriminative models. Phd thesis, Aachen, Germany, 2008. (Cité en page 84.)
- [DeSouza 2002] Guilherme N. DeSouza et Avinash C. Kak. *Vision for Mobile Robot Navigation : A Survey*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, pages 237–267, 2002. (Cité en page 2.)
- [Dominique 1995] Attali Dominique. *Squelettes et graphes de Voronoi 2D et 3D*. Thèse de doctorat, 1995. (Cité en page 25.)
- [Dupont 2003] Florent Dupont, Benjamin Gilles et Atilla Baskurt. *Progressive transmission of 3D object based on balls and cones union from medial axis transformation*. In ICIP, pages 797–800, 2003. (Cité en page 29.)
- [Ea 2001] Thomas Ea, Olivier ROMAIN, Claude GASTAUD et Patrick GARDA. *Un capteur de sphéro-vision stéréoscopique couleur*. In ORASIS 2001, Congrès francophone de vision, Cahors, 5-8 juin 2001, pages 301–308, IRIT, 118, route de Narbonne, 31062 Toulouse Cedex 4, France, Juin 2001. (Cité en pages 8 et 9.)
- [Eede 2006] M. Van Eede, Diego Macrini, Alexandru Telea, Cristian Sminchisescu et Sven J. Dickinson. *Canonical Skeletons for Shape Matching*. In ICPR (2), pages 64–69, 2006. (Cité en page 29.)
- [Elfes 1987] Alberto Elfes. *Sonar-based real-world mapping and navigation*. International Journal of Robotics and Automation, 1987. (Cité en page 96.)
- [Ellekilde 2007] Lars-Peter Ellekilde, Shoudong Huang, Jaime Valls Miró et Gamin Dissanayake. *Dense 3D Map Construction for Indoor Search and Rescue*. J. Field Robotics, pages 71–89, 2007. (Cité en page 97.)
- [Fabrizi 2000] Elisabetta Fabrizi et Alessandro Saffiotti. *Extracting Topology-Based Maps from Gridmaps*. In ICRA, pages 2972–2978, 2000. (Cité en page 97.)
- [Farhan 2011] Hazim A. Farhan, Hussein H. Owaied et Suhaib I. Al-Ghazi. *Finding Shortest Path for Developed Cognitive Map Using Medial Axis*. CoRR, 2011. (Cité en page 30.)
- [Garrido 2011] Santiago Garrido, Luis Moreno, Dolores Blanco et Piotr Jurewitz. *Path planning for mobile robot navigation using Voronoi diagram and fast*

- marching*. International Journal of Robotics and Automation, pages 154–176, 2011. (Cité en page 30.)
- [Ge 1996] Yaorong Ge et J. Michael Fitzpatrick. *On the Generation of Skeletons from Discrete Euclidean Distance Maps*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 11, pages 1055–1066, 1996. (Cité en page 27.)
- [Geyer 2001] C. Geyer et K. Daniilidis. *Catadioptric projective geometry*. IJCV, vol. 45, no. 3, pages 223–243, 2001. (Cité en page 8.)
- [Giesen 2009] Joachim Giesen, Balint Miklos, Mark Pauly et Camille Wormser. *The scale axis transform*. In Symposium on Computational Geometry, pages 106–115, 2009. (Cité en page 29.)
- [Goedeme 2004] Toon Goedeme, T. Tuytelaars et L. Van Gool. *Fast wide baseline matching for visual navigation*. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 1, pages I–24–I–29 Vol.1, 2004. (Cité en page 63.)
- [Gonzalez-Barbosa 2002] José-Joel Gonzalez-Barbosa et Simon Lacroix. *Rover Localization in Natural Environments by Indexing Panoramic Images*. In ICRA, pages 1365–1370, 2002. (Cité en page 64.)
- [Halawani 2004] Alaa Halawani et Hans Burkhardt. *Image Retrieval by Local Evaluation of Nonlinear Kernel Functions around Salient Points*. In ICPR (2), pages 955–960, 2004. (Cité en page 65.)
- [Hesslink 2007] W. Hesslink. *A linear-time algorithm for Euclidean feature transform sets*. Inf. Process. Lett., vol. 102, no. 5, pages 181–186, 2007. (Cité en pages 26, 27, 28, 29, 31, 32, 33, 34, 38, 40, 41 et 43.)
- [Hirata 1996] T. Hirata. *A unified linear-time algorithm for computing distance maps*. Inf. Process. Lett., vol. 58, no. 3, pages 129–133, 1996. (Cité en pages 27, 33, 34 et 51.)
- [Ishiguro 1996] Hiroshi Ishiguro et Saburo Tsuji. *Image-based memory of environment*. In IROS, pages 634–639, 1996. (Cité en page 64.)
- [Kegl 2002] B. Kegl et A. Krzyzak. *Piecewise linear skeletonization using principal curves*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 1, pages 59–74, 2002. (Cité en page 29.)
- [Kimmel 1995] Ron Kimmel, Doron Shaked, Nahum Kiryati et Alfred M. Bruckstein. *Skeletonization via Distance Maps and Level Sets*. Computer Vision and Image Understanding, vol. 62, pages 382–391, 1995. (Cité en page 27.)
- [Kosecka 2004] Jana Kosecka et Fayin Li. *Vision based Topological Markov Localization*. In ICRA, pages 1481–1486, 2004. (Cité en page 64.)
- [Kuipers 2000] Benjamin Kuipers. *The Spatial Semantic Hierarchy*. Artif. Intell., pages 191–233, 2000. (Cité en page 97.)
- [Kuipers 2004] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon et Francesco Savelli. *Local Metrical and Global Topological Maps in the*

- Hybrid Spatial Semantic Hierarchy*. In ICRA, pages 4845–4851, 2004. (Cité en page 97.)
- [Labbani-Igbida 2011] Ouiddad Labbani-Igbida, Cyril Charron et El Mustapha Mouaddib. *Haar invariant signatures and spatial recognition using omnidirectional visual information only*. *Autonomous Robots*, vol. 30, no. 3, pages 333–349, 2011. (Cité en pages 65 et 66.)
- [Labrosse 2006] Frédéric Labrosse. *The visual compass : Performance and limitations of an appearance-based method*. *J. Field Robotics*, vol. 23, no. 10, pages 913–941, 2006. (Cité en page 113.)
- [Lam 1992] Louisa Lam, Seong-Whan Lee et Ching Y. Suen. *Thinning Methodologies-A Comprehensive Survey*. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 869–885, 1992. (Cité en page 26.)
- [Leonard 1991] John J. Leonard et Hugh F. Durrant-Whyte. *Mobile robot localization by tracking geometric beacons*. *IEEE T. Robotics and Automation*, vol. 7, no. 3, pages 376–382, 1991. (Cité en page 2.)
- [Leymarie 1992] Frederic F. Leymarie et Martin D. Levine. *Simulating the Grassfire Transform Using an Active Contour Model*. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 1, pages 56–75, 1992. (Cité en page 27.)
- [Li 2013] You Li et Y. Ruichek. *Building variable resolution occupancy grid map from stereoscopic system - A quadtree based approach*. In *Intelligent Vehicles Symposium (IV)*, 2013 IEEE, 2013. (Cité en page 96.)
- [Liu 2011] Lu Liu, Erin W. Chambers, David Letscher et Tao Ju. *Extended grassfire transform on medial axes of 2D shapes*. *Comput. Aided Des.*, vol. 43, no. 11, pages 1496–1505, 2011. (Cité en page 26.)
- [Lowe 1999] David G. Lowe. *Object Recognition from Local Scale-Invariant Features*. In *ICCV*, pages 1150–1157, 1999. (Cité en page 63.)
- [Lowe 2004] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. *International Journal of Computer Vision*, vol. 60, no. 2, 2004. (Cité en page 63.)
- [Lozenguez 2011] Guillaume Lozenguez, Lounis Adouane, Aurélie Beynier, Philippe Martinet et Abdel-Ilhah Mouaddib. *Map Partitioning to Approximate an Exploration Strategy in Mobile Robotics*. In *PAAMS*, pages 63–72, 2011. (Cité en pages 43 et 46.)
- [Lozenguez 2012] Guillaume Lozenguez, Lounis Adouane, Aurélie Beynier, Abdel-Ilhah Mouaddib et Philippe Martinet. *Map partitioning to approximate an exploration strategy in mobile robotics*. *Multiagent and Grid Systems*, vol. 8, no. 3, pages 275–288, 2012. (Cité en pages 43 et 46.)
- [Malandain 1998] Grégoire Malandain et Sara Fernández-Vidal. *Euclidean skeletons*. *Image Vision Comput.*, vol. 16, no. 5, pages 317–327, 1998. (Cité en page 29.)



- [Marie 2013a] Romain Marie, Ouiddad Labbani-Igbida, Pauline Merveilleux et El Mustapha Mouaddib. *Autonomous robot exploration and cognitive map building in unknown environments using omnidirectional visual information only*. Workshop On Robotic Vision, 2013. (Cité en page 98.)
- [Marie 2013b] Romain Marie, Ouiddad Labbani-Igbida, El Mustapha Mouaddib et al. *Exploration autonome et construction de carte cognitive en environnement inconnu basées vision omnidirectionnelle uniquement*. In Orais, Congrès des jeunes chercheurs en vision par ordinateur, 2013. (Cité en page 98.)
- [Marie 2013c] Romain Marie, Ouiddad Labbani-Igbida et Mustapha Mouaddib. *The delta-medial axis : a robust and linear time algorithm for euclidian skeleton computation*. In International Conference on Image Processing, 2013. (Cité en pages 22, 40 et 41.)
- [Masehian 2003] E. Masehian, M.R. Amin-Naseri et S.E. Khadem. *Online motion planning using incremental construction of medial axis*. In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, volume 3, pages 2928–2933 vol.3, 2003. (Cité en page 30.)
- [Mat 1988] *Examples of topological properties of skeletons*. Image Analysis and Mathematical Morphology, Theoretical advances, vol. 2, 1988. (Cité en page 24.)
- [Matas 2002] J. Matas, O. Chum, M. Urban et T. Pajdla. *Robust Wide Baseline Stereo from Maximally Stable Extremal Regions*. In Proceedings of the British Machine Vision Conference, pages 36.1–36.10, 2002. (Cité en page 63.)
- [Mattioli 1993] J. Mattioli. *Problèmes inverses et relations différentielles en morphologie mathématique*. Thèse, Université Paris Dauphine, 1993. (Cité en page 24.)
- [Mayya 1996] Niranjana Mayya et V. T. Rajan. *Voronoi diagrams of polygons : A framework for shape representation*. Journal of Mathematical Imaging and Vision, vol. 6, no. 4, pages 355–378, 1996. (Cité en page 27.)
- [Meijster 2000] A. Meijster, J. B. T. M. Roerdink et W. H. Hesselink. *A General Algorithm For Computing Distance Transforms In Linear Time*, 2000. (Cité en page 34.)
- [Menegatti 2004] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello et Hiroshi Ishiguro. *Image-based Monte Carlo localisation with omnidirectional images*. Robotics and Autonomous Systems, vol. 48, no. 1, pages 17–30, 2004. (Cité en page 64.)
- [Merveilleux 2010] Pauline Merveilleux, Ouiddad Labbani-Igbida et El Mustapha Mouaddib. *Free space detection using active contours in omnidirectional images*. In ICIP, pages 3533–3536, 2010. (Cité en page 14.)
- [Merveilleux 2011a] Pauline Merveilleux, Ouiddad Labbani-Igbida et El Mustapha Mouaddib. *Real-time free space detection and navigation using omnidirectional vision and parametric and geometric active contours*. In ICRA, pages 6312–6317, 2011. (Cité en page 14.)

- [Merveilleux 2011b] Pauline Merveilleux, Ouidad Labbani-Igbida et El Mustapha Mouaddib. *Robust free space segmentation using active contours and monocular omnidirectional vision*. In ICIP, pages 2877–2880, 2011. (Cité en page 14.)
- [Merveilleux 2013] Pauline Merveilleux, Ouidad Labbani-Igbida et El Mustapha Mouaddib. *Robust free space segmentation using active contours and monocular omnidirectional vision*. In Thèse, Université de Picardie Jules Verne, 2013. (Cité en pages 8, 14, 15, 16, 17, 18 et 96.)
- [Meyer 1979] Fernand Meyer. *Cytologie quantitative et morphologie mathématique*. Thèse, Ecole des Mines de Paris, 1979. (Cité en page 29.)
- [Miklos 2010] Balint Miklos, Joachim Giesen et Mark Pauly. *Discrete scale axis representations for 3D geometry*. ACM Trans. Graph., vol. 29, no. 4, 2010. (Cité en page 29.)
- [Moayer 1975] Bijan Moayer et King-Sun Fu. *A syntactic approach to fingerprint pattern recognition*. Pattern Recognition, vol. 7, no. 1-2, pages 1–23, 1975. (Cité en page 29.)
- [Mokhtarian 1992] Farzin Mokhtarian et Alan K. Mackworth. *A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 8, pages 789–805, 1992. (Cité en page 28.)
- [Mouaddib 2005] E Mouaddib. *Introduction à la Vision Panoramique Catadioptrique*. Traitement du Signal, vol. 22, September 2005. (Cité en page 8.)
- [Mozos 2006] Óscar Martínez Mozos et Wolfram Burgard. *Supervised Learning of Topological Maps using Semantic Information Extracted from Range Data*. In IROS, pages 2772–2777, 2006. (Cité en page 97.)
- [Murillo 2009] A. C. Murillo et J. Kosecka, 2009. (Cité en page 65.)
- [Nguyen 2012] Thien-Nghia Nguyen, Bernd Michaelis, Ayoub Al-Hamadi, Michael Tornow et Marc-Michael Meinecke. *Stereo-Camera-Based Urban Environment Perception Using Occupancy Grid and Object Tracking*. IEEE Transactions on Intelligent Transportation Systems, pages 154–165, 2012. (Cité en page 96.)
- [Ogniewicz 1995] Robert L. Ogniewicz et Olaf Kübler. *Hierarchical Voronoi skeletons*. Pattern Recognition, vol. 28, pages 343–359, 1995. (Cité en page 27.)
- [Oliva 2001] Aude Oliva et Antonio Torralba. *Modeling the Shape of the Scene : A Holistic Representation of the Spatial Envelope*. International Journal of Computer Vision, vol. 42, no. 3, pages 145–175, 2001. (Cité en page 64.)
- [Oliva 2006] Aude Oliva et Antonio Torralba. *Building the gist of a scene : the role of global image features in recognition*. In Progress in Brain Research, 2006. (Cité en page 65.)
- [Pajdla 1999] Tomáš Pajdla et Václav Hlavác. *Zero Phase Representation of Panoramic Images for Image Vased Localization*. In CAIP, pages 550–557, 1999. (Cité en page 64.)

- [Pfaltz 1967] John L. Pfaltz et Azriel Rosenfeld. *Computer representation of planar regions by their skeletons*. Commun. ACM, vol. 10, no. 2, pages 119–122, 1967. (Cit  en pages 24 et 25.)
- [Pizer 1987] Stephen M. Pizer, William R. Oliver et Sandra H. Bloomberg. *Hierarchical Shape Description Via the Multiresolution Symmetric Axis Transform*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 9, no. 4, pages 505–511, 1987. (Cit  en page 28.)
- [Postolski 2013] Michal Postolski, Michel Couprie et Marcin Janaszewski. *Scale Filtered Euclidean Medial Axis*. In DGCI, pages 360–371, 2013. (Cit  en page 29.)
- [Pugeault 2010] Nicolas Pugeault et Richard Bowden. *Learning Pre-attentive Driving Behaviour from Holistic Visual Features*. In ECCV (6), pages 154–167, 2010. (Cit  en page 65.)
- [Pujar 2010] Jagadish H. Pujar et Pallavi S. Gurjal. *Binary Data Compression Using Medial Axis Transform Algorithm*. volume 70 of *Communications in Computer and Information Science*. Springer, 2010. (Cit  en page 29.)
- [Ramisa 2009] Arnau Ramisa, Adriana Tapus, David Aldavert, Ricardo Toledo et Ramon Lopez De Mantaras. *Robust Mobile Robot Localization using Panoramic Vision and Combinations of Local Feature Region Detectors*. Autonomous Robots, vol. 27, pages 373–385, 2009. (Cit  en page 63.)
- [Riley 1964] J. Riley et L. Calabi. *Certain properties of circles inscribed in simple closed curves*. Technical report 59281, Park Math. Lab. Inc, 1964. (Cit  en page 24.)
- [Romain 2001] Olivier Romain, Thomas EA, Claude GASTAUD et Patrick GARDA. *Un capteur multi-spectral de vision panoramique 3D*. In ORASIS 2001, Congr s francophone de vision, Cahors, 5-8 juin 2001, pages 359–366, IRIT, 118, route de Narbonne, 31062 Toulouse Cedex 4, France, Juin 2001. (Cit  en pages 8 et 9.)
- [Ronse 1988] Christian Ronse. *Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images*. Discrete Applied Mathematics, vol. 21, no. 1, pages 67–79, 1988. (Cit  en page 26.)
- [Rubner 2000] Yossi Rubner, Carlo Tomasi et LeonidasJ. Guibas. *The Earth Mover’s Distance as a Metric for Image Retrieval*. International Journal of Computer Vision, vol. 40, no. 2, pages 99–121, 2000. (Cit  en page 81.)
- [Rumpf 2002] Martin Rumpf et Tobias Preu er. *A Level Set Method for Anisotropic Geometric Diffusion in 3D Image Processing*. SIAM Journal of Applied Mathematics, vol. 62, no. 5, 2002. (Cit  en pages 27 et 28.)
- [Scaramuzza 2008] Davide Scaramuzza, C dric Pradalier et Roland Siegwart. *Performance evaluation of a vertical line descriptor for omnidirectional images*. In IROS, pages 3127–3132, 2008. (Cit  en page 63.)

- [Scaramuzza 2011a] Davide Scaramuzza. *Performance evaluation of 1-point-RANSAC visual odometry*. J. Field Robotics, vol. 28, no. 5, pages 792–811, 2011. (Cité en page 101.)
- [Scaramuzza 2011b] Davide Scaramuzza et Friedrich Fraundorfer. *Visual Odometry [Tutorial]*. IEEE Robot. Automat. Mag., vol. 18, no. 4, pages 80–92, 2011. (Cité en page 101.)
- [Schmitt 1993] M. Schmitt et J. Mattioli. *Morphologie mathématique*. Masson, 1993. (Cité en page 24.)
- [Schulz-mirbach 1996] H. Schulz-mirbach, H. Burkhardt, S. Siggelkow et Technische Informatik I. *Using Invariant Features For Content Based Data Retrieval*, 1996. (Cité en pages 65 et 66.)
- [Se 2002] Stephen Se, David G. Lowe et James J. Little. *Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks*. I. J. Robotic Res., vol. 21, no. 8, pages 735–760, 2002. (Cité en page 63.)
- [Sebastian 2001] Thomas B. Sebastian, Philip N. Klein et Benjamin B. Kimia. *Recognition of Shapes by Editing Shock Graphs*. In ICCV, pages 755–762, 2001. (Cité en pages 38 et 42.)
- [Sebastian 2005] Thomas B. Sebastian et Benjamin B. Kimia. *Curves vs. skeletons in object recognition*. Signal Processing, vol. 85, no. 2, pages 247–263, 2005. (Cité en page 29.)
- [She 2009] F. H. She, R. H. Chen, W. M. Gao, P. H. Hodgson, L. X. Kong et H. Y. Hong. *Improved 3D Thinning Algorithms for Skeleton Extraction*. In DICTA, pages 14–18. IEEE Computer Society, 2009. (Cité en page 26.)
- [Siagian 2007] Christian Siagian et Laurent Itti. *Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 2, pages 300–312, 2007. (Cité en page 65.)
- [Siddiqi 1998] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson et Steven W. Zucker. *Shock Graphs and Shape Matching*, 1998. (Cité en page 29.)
- [Siddiqi 2002] Kaleem Siddiqi, Sylvain Bouix, Allen Tannenbaum et Steven W. Zucker. *Hamilton-Jacobi Skeletons*. International Journal of Computer Vision, vol. 48, no. 3, pages 215–231, 2002. (Cité en page 27.)
- [Siddiqi 2008] Kaleem Siddiqi et Stephen Pizer. *Medial representations : Mathematics, algorithms and applications*. Springer Publishing Company, Incorporated, 1st édition, 2008. (Cité en page 26.)
- [Siggelkow 2002] Sven Siggelkow. *Feature histograms for content-based image retrieval*. PhD thesis, 2002. (Cité en pages 65, 66 et 76.)
- [Singh 2010] Gautam Singh et J Kosecka. *Visual loop closing using gist descriptors in manhattan world*. International Conference on Robotics and Automation, 2010. (Cité en page 65.)

- [Talbot 1992] Hugues Talbot. *Euclidean skeletons and conditional bisectors*. In in Visual Communications and Image Processing'92, pages 862–876, 1992. (Cité en page 29.)
- [Tamimi 2006] Hashem Tamimi, Henrik Andreasson, André Treptow, Tom Duckett et Andreas Zell. *Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT*. Robotics and Autonomous Systems, vol. 54, no. 9, 2006. (Cité en page 63.)
- [Tapus 2006] Adriana Tapus et Roland Siegwart. *A Cognitive Modeling of Space using Fingerprints of Places for Mobile Robot Navigation*. In ICRA, pages 1188–1193, 2006. (Cité en page 97.)
- [Thrun 2003] Sebastian Thrun. *Learning Occupancy Grid Maps with Forward Sensor Models*. Autonomous Robots, 2003. (Cité en page 96.)
- [Torralba 2003] Antonio Torralba. *Contextual Priming for Object Detection*. International Journal of Computer Vision, vol. 53, no. 2, pages 169–191, 2003. (Cité en page 65.)
- [Tuytelaars 2004] Tinne Tuytelaars et Luc Van Gool. *Matching Widely Separated Views Based on Affine Invariant Regions*. International Journal of Computer Vision, vol. 59, no. 1, pages 61–85, 2004. (Cité en page 63.)
- [Ulrich 2000] Iwan Ulrich et Illah R. Nourbakhsh. *Appearance-Based Place Recognition for Topological Localization*. In ICRA, pages 1023–1029, 2000. (Cité en pages 64 et 97.)
- [Valgren 2006] Christoffer Valgren, Achim J. Lilienthal et Tom Duckett. *Incremental Topological Mapping Using Omnidirectional Vision*. In IROS, pages 3441–3447, 2006. (Cité en page 63.)
- [Valgren 2010] Christoffer Valgren et Achim J. Lilienthal. *SIFT, SURF & seasons : Appearance-based long-term localization in outdoor environments*. Robotics and Autonomous Systems, vol. 58, no. 2, pages 149–156, 2010. (Cité en page 63.)
- [Victorino 2003] Alessandro Corrêa Victorino, Patrick Rives et Jean-Jacques Borrelly. *Safe navigation for indoor mobile robots. Part I : A sensor-based navigation framework*. International Journal of Robotics Research, vol. 22, no. 12, 2003. (Cité en page 114.)
- [Wolf 2002] J. Wolf, W. Burgard et H. Burkhardt. *Using an Image Retrieval System for Vision-based Mobile Robot Localization*. In Proc. of the International Conference on Image and Video Retrieval (CIVR), 2002. (Cité en pages 65 et 66.)
- [Wolter 1992] Franzerich Wolter. *Cut locus and medial axis in global shape interrogation and representation*. In MIT Design Laboratory Memorandum 92-2 and MIT Sea Grant Report, 1992. (Cité en page 24.)
- [Zhou 1995] R.W. Zhou, C. Quek et G.S. Ng. *A novel single-pass thinning algorithm and an effective set of performance criteria*. Pattern Recognition Letters, vol. 16, no. 12, pages 1267 – 1275, 1995. (Cité en page 26.)

- 
- [Zhou 2003] Chao Zhou, Yucheng Wei et Tieniu Tan. *Mobile Robot Self-Localization Based on Global Visual Appearance Features*. In Proceedings of the 2003 IEEE Int. Conf. on Robotics and Automation, ICRA 2003, September 14-19, 2003, pages 1271–1276, 2003. (Cité en page 64.)



