Robotique coopérative aéro-terrestre : Localisation et cartographie hétérogène

Brice Renaudeau

▶ **To cite this version:**

Brice Renaudeau. Robotique coopérative aéro-terrestre : Localisation et cartographie hétérogène. Automatique / Robotique. Université de Limoges, 2019. Français. NNT : 2019LIMO0012 . tel-02096377

HAL Id: tel-02096377

https://theses.hal.science/tel-02096377

Submitted on 11 Apr 2019

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.

Thèse de doctorat

Université de Limoges

Université de Limoges
ED 610 - Sciences et Ingénierie des Systèmes, Mathématiques, Informatique (SISMI)
Xlim

Thèse pour obtenir le grade de
Docteur de l'Université de Limoges
Robotique

Présentée et soutenue par
Brice Renaudeau

Le 7 mars 2019

Robotique coopérative aéro-terrestre : localisation et cartographie hétérogène

Thèse dirigée par Ouiddad Labbani-Igbida et Gilles Mourioux

JURY :

Président du jury

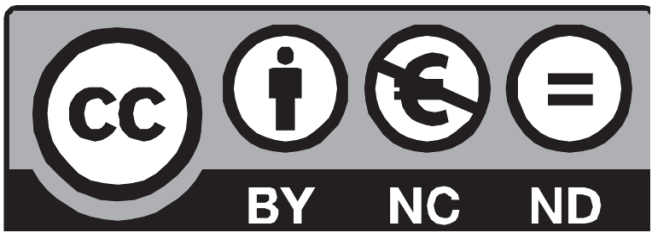
M. David Filliat, Professeur, ENSTA-ParisTech

Rapporteurs
M. Simon Lacroix, Directeur de recherche , LAAS/CNRS,
M. Youcef Mezouar, Professeur, SIGMA-Clermont, Institut Pascal

Examinateurs
M. Guillaume Caron, Maître de conférences, Université de Picardie Jules Verne

Licence CC BY-NC-ND 3.0

Cette création est mise à disposition selon le Contrat :
«**Attribution-Pas d'Utilisation Commerciale-Pas de modification 3.0 France**»
disponible en ligne : *http ://creativecommons.org/licenses/by-nc-nd/3.0/fr/*



Résumé

Les travaux de cette thèse adressent la problématique de la coopération aéro-terrestre pour la cartographie de l'espace navigable. La nécessité d'une carte pour la navigation et la planification de chemins pour les robots terrestres n'est plus à prouver. L'utilisation d'une coopération aéro-terrestre pour créer une carte navigable à destination du robot terrestre a plusieurs intérêts. Premièrement, le drone peut cartographier rapidement une zone grâce à son champ de vision étendu et ses capacités de déplacement. Deuxièmement, la fusion des cartes créées par ces deux agents permet de tirer le meilleur profit des deux points de vue : la cohérence de la vue aérienne globale et la précision de la vue terrestre locale.

Pour répondre à cette problématique, nous proposons une méthode qui s'appuie sur la création de cartes hybrides et leur fusion. Les cartes sont construites en utilisant le squelette de l'espace navigable terrestre comme support d'un graphe contenant également des informations métriques locales de l'environnement. La mise en correspondance des cartes aérienne et terrestre s'effectue à l'aide d'un appariement point à point déterminé grâce à une mesure de dissimilarité appropriée. Cette dernière est définie pour répondre aux critères d'invariance et de discriminance dans ce contexte. La mise en correspondance est ensuite utilisée pour fusionner les cartes entre elles.

Les cartes fusionnées peuvent être utilisées par le robot au sol pour effectuer sa mission. Elles permettent également de propager des informations telles que des coordonnées GPS à des robots et dans des lieux où ce dispositif n'est pas disponible. Des expérimentations en environnements virtuels et réels sont réalisées pour valider cette approche et en tracer les perspectives.

Mots clés : Coopération mutli-robots, SLAM, Cartographie, Systèmes aéro-terrestre, traversabilité

Licence CC BY-NC-ND 3.0

Summary

This work aims to study the problem of air-ground robotic cooperation for collaborative traversability mapping. The need for a map for navigation and path planning for terrestrial robots is no longer to be proven. The use of air-ground cooperation to create a navigable map for the ground robots has several interests. First, the drone can quickly map an area through its large field of vision and traveling capabilities. Second, the fusion of maps based on these two agents makes it possible to draw the best benefits from both points of views : the coherence of the global aerial view and the accuracy of the local ground view.

To answer this problem, we propose a method that relies on the construction of a unified model of hybrid maps and their fusion.The maps are built using the skeleton of the traversability space as a support for graphs also containing local metric and potentialy semantic information of the environment. The maching of aerial and ground maps is done using a point to point correlation based on an appropriate dissimilarity measure. This measure is defined to meet invariance and discriminance criteria. The matching is then used to merge the maps into an augmented traversability map.

The merged maps can be used by the ground robot to perform its mission. They also make it possible to deploy information such as GPS coordinates to robots in GPS denied environments. Experiments in virtual and real world environments have been carried out to validate this approach and map out future perspetives.

Keywords : Multirobot cooperation, SLAM, Mapping, Air-ground systems, Traversability mapping

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Table des matières

Résumé **5**

Liste des symboles **5**

Introduction générale **7**

1. État de l'art de la cartographie multirobots **11**
1.1. Localisation et cartographie simultanées 13
1.1.1. Optimisation conjointe 14
1.1.2. Pré-traitement dépendant du capteur 16
1.2. Représentation de l'environnement 18
1.2.1. Métrique : la géométrie de l'environnement 19
1.2.2. Topologique : les connexions entre lieux 22
1.2.3. Hybride : la complémentarité des deux approches 25
1.2.4. Récapitulatif 28
1.3. Coopération multirobots 29
1.3.1. Système multirobots homogène ou hétérogène 30
1.3.2. Architecture centralisée, décentralisée ou distribuée 32
1.3.3. Planification externe ou autonome 34
1.4. Les défis à relever 35
Conclusion 37

2. Cadre de l'étude et représentation hétérogène **39**
Introduction 40
2.1. Travaux préliminaires 40
2.1.1. Détection de l'espace navigable 41
2.1.2. Squelettisation de l'espace navigable 42
2.1.3. Création d'un graphe à partir d'un squelette 46
2.2. Cadre de l'étude et objectifs 47
2.2.1. Limites de l'exploration terrestre 47
2.2.2. Objectifs d'une cartographie multirobots 48
2.2.3. Représentation topologique modulaire et évolutive 48
2.3. Définition de la représentation hybride 49
2.3.1. Structure de graphe hybride 49
2.3.2. Architecture logicielle 52
2.4. Carte dépendante de la source 54
2.4.1. Formalisation de l'espace navigable 54

2.4.2. Carte aérienne . . . 55
2.4.3. Carte terrestre . . . 56
2.4.4. Carte satellite . . . 58
Conclusion . . . 59

3. Fusion de cartes aéro-terrestres 61
Introduction . . . 62
3.1. Appariement point à point . . . 62
3.2. Descripteur et mesure de dissimilarité . . . 65
3.2.1. Invariance, discriminance et complétude . . . 65
3.2.2. Descripteur, structuration des informations utiles . . . 66
3.2.3. Mesure de dissimilarité . . . 67
3.3. Algorithmes de Matching Hiérarchique . . . 70
3.3.1. Algorithme Hongrois . . . 70
3.3.2. Mesure « raffinée » . . . 72
3.3.3. Suppression des assignations erronées . . . 75
3.3.4. Matching topologique . . . 76
3.3.5. Assignation par consensus de projection . . . 77
3.3.6. Résultat du matching . . . 78
3.3.7. Validation du matching . . . 80
3.4. Analyse et discussion de l'approche proposée . . . 81
3.4.1. Analyse de résultats . . . 81
3.4.2. Méthode d'optimisation alternative . . . 83
3.4.3. Conclusion . . . 84
3.5. Fusion des cartes . . . 85
3.5.1. Stratégie adaptée au contenu des cartes . . . 86
3.5.2. Carte augmentée construite par projection . . . 86
3.6. Utilisation de la carte fusionnée . . . 88
Conclusion . . . 90

4. Évaluations expérimentales 91
Introduction . . . 92
4.1. Présentation des sites d'expérimentations . . . 92
4.1.1. Plateforme expérimentale PAVIN . . . 92
4.1.2. Parking à proximité du laboratoire . . . 94
4.1.3. Hall d'essais de l'ENSIL-ENSCI . . . 95
4.1.4. Étage de l'ENSIL-ENSCI . . . 95
4.1.5. Expérimentations à l'ETH de Zurich . . . 96
4.1.6. Environnements de développement . . . 97
4.2. Création de cartes aériennes . . . 98
4.2.1. Détection de l'espace navigable . . . 98
4.2.2. Gestion de la visibilité partielle de l'environnement . . . 102
4.2.3. Gestion des éléments dynamiques . . . 105

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

4.3. Évaluation de la cartographie coopérative . . . 108
4.3.1. Impact des paramètres de la mesure . . . 108
4.3.2. Évaluation de la mise en correspondance des cartes . . . 116
4.3.3. Localisation à partir de la carte fusionnée . . . 120
4.4. Utilisation de cartes en ligne . . . 122
4.4.1. Création de cartes géoréférencées . . . 123
4.4.2. Matching des cartes OSM et aérienne . . . 124
Discussion . . . 127

5. Conclusion et perspectives **129**
5.1. Bilan Général . . . 129
5.2. Pour aller plus loin . . . 130

A. Téléchargement des cartes OpenStreetMap **133**
A.1. API . . . 134
A.2. Fichier .osm . . . 134
A.3. Filtrage . . . 137
A.4. Projection dans le repère local tangent . . . 137

Bibliographie **141**

Licence CC BY-NC-ND 3.0

Table des figures

1.1. Structure générale d'un algorithme de SLAM. 13
1.2. Représentation du problème de SLAM sous forme de *factor graph*. . . 14
1.3. Correction de la carte et de la position grâce à une fermeture de boucle. 17
1.4. Les 3 grands formats de représentation 2D. 19
1.5. Cartes géométriques : a) Lidar 2D b) Points et lignes 3D. 20
1.6. Cartes discrètes : a) Grille 2D, b) MNT 2.5D, c)Voxels 3D. 21
1.7. Carte topologique d'un bâtiment. 22
1.8. Différents types de représentations topologiques. 24
1.9. Superposition de représentation métriques et topologiques. 26
1.10. Assemblage de cartes locales. 27
1.11. Hiérarchie des représentations. 28
1.12. Photo de la Robocup (Small Size League). 31
1.13. Assemblage de robots homogènes. 31
1.14. Chalenge européen euRathlon. 32
1.15. Les trois types d'architectures : centralisée, décentralisée et distribuée. 33
1.16. Projets européens de robotique hétérogène. 34

2.1. Capteur catadioptrique et image omnidirectionnelle, et robot Pioneer. 40
2.2. Images omnidirectionnelles avec le contour actif de l'espace navigable. 41
2.3. Espace navigable calculé à partir d'un plan d'un bâtiment. 42
2.4. Exemple d'axe médian pour une image de raie. 42
2.5. Carte des distances d'une image binaire. 44
2.6. Illustration de la condition du DMA. 46
2.7. Illustration de l'axe médian pour une forme. 46
2.8. Graphe créé à partir de l'axe médian. 47
2.9. Représentation hybride proposée. 50
2.10. Carte hybride topométrique d'un environnement intérieur. 51
2.11. Données sémantiques portées par la carte hybride. 51
2.12. Diagramme de l'architecture logicielle. 52
2.13. Cartographie 3D avec la trajectoire du drone. 53
2.14. Segmentation et cartographie de l'image aérienne. 55
2.15. Carte globale du site Pavin créée par le drone. 56
2.16. Cartographie terrestre locale par vision catadioptrique. 56
2.17. Carte du site Pavin créée par le robot au sol. 57
2.18. Carte hybride construite à partir des données de OpenStreetMap. . . 58

3.1. Présentation du matching de nœuds. 63

3.2. Représentation du problème sous forme de carte de couts. 65
3.3. Illustration du descripteur d'un nœud. 66
3.4. Deux représentations différentes d'un même lieu géographique. 67
3.5. Mesure de différence entre deux nœuds pour la même intersection. . . 68
3.6. Résultat de l'algorithme Hongrois utilisant la mesure initiale. 71
3.7. Carte de couts assignée avec la méthode Hongroise. 72
3.8. Résultat de l'algorithme Hongrois utilisant la mesure raffinée. 74
3.9. Évolution de la carte de couts et de l'assignation. 74
3.10. Association implicite des arêtes et détection de faux matching. . . . 75
3.11. Association de nœuds en utilisant les voisins matchés. 76
3.12. Résultat du matching topologique. 77
3.13. Carte de couts après matching par projection. 78
3.14. Architecture du matching hiérarchique. 78
3.15. Résultat du matching hiérarchique aéro-terrestre. 78
3.16. Logigramme de l'algorithme de matching de graphes. 79
3.17. Diagramme du score de matching. 82
3.18. Courbe de précision/rappel du score de matching. 82
3.19. Temps de calcul de l'algorithme. 83
3.20. Représentations topologiques différentes de la même intersection. . . 85
3.21. Carte augmentée obtenue par fusion des cartes aérienne et terrestre. 87
3.22. Localisation et navigation terrestre à l'aide de la carte fusionnée. . . 88

4.1. Site expérimental PAVIN. 93
4.2. Modèle 3D du site expérimental PAVIN. 93
4.3. Vue de drone du parking. 94
4.4. Drone *Bebop 2* (*Parrot*), utilisé dans les expérimentations. 94
4.5. Environnement de test de la coopération aéro-terrestre. 95
4.6. Plan de l'étage de l'ENSIL-ENSCI. 96
4.7. Espace vert à Zurich et bâtiment à Genève. 96
4.8. Interface graphique du simulateur 4DVirtualiz. 98
4.9. Transformation en orthoimage. 99
4.10. Image aérienne provenant du simulateur et segmentation couleur. . . 99
4.11. Normales au sol permettant de déterminer la traversabilité. 100
4.12. Carte 3D du bâtiment et l'espace navigable estimé. 101
4.13. Grille d'élévation avec l'espace navigable en transparence. 101
4.14. Variation de la carte hybride d'un même lieu. 102
4.15. Mosaïque d'images appliquée au site expérimental PAVIN. 103
4.16. Matching de cartes aériennes locales. 104
4.17. Indice de confiance des nœuds. 105
4.18. Véhicules mobiles lors de la cartographie. 106
4.19. Logigramme de l'algorithme de matching de graphes. 107
4.20. Cartes de couts en fonction du pas d'échantillonnage de la rotation. . 110
4.21. Valeur de dissimilarité en fonction de l'angle α. 111
4.22. Zoom sur la zone en dessous du seuil C_{max}. 111

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

4.23. Carte de couts en fonction de K_d. 112
4.24. Carte de couts en fonction de K_c. 113
4.25. Carte de couts en fonction de K_a. 114
4.26. Carte de couts en fonction de K_λ. 115
4.27. Carte de couts en fonction de K_α. 116
4.28. Trajectoires effectuées par les robots dans PAVIN. 117
4.29. Carte aérienne avec représentation multiple. 119
4.30. Cartes terrestres avec des représentations différentes du même lieu. . 119
4.31. Fusion de cartes avec des représentations multiples. 119
4.32. Fusion des cartes aéro-terrestres dans un environnement intérieur. . . 120
4.33. Re-localisation du robot à l'aide de la carte augmentée. 121
4.34. Évolution de la covariance sur la localisation du robot 122
4.35. Carte hybride du site Pavin construite avec OpenStreetMap. 123
4.36. Différence de représentations entre la carte OSM et l'axe médian. . . 124
4.37. Matching entre la carte routière et la carte créée par le drone. 125
4.38. Projection de la carte aérienne dans le repère de la carte OSM. 126

A.1. Logigramme du téléchargement de la carte OpenStreetMap. 133
A.2. Interface web permettant de sélectionner la zone à télécharger. 134
A.3. Format des fichiers .osm. 135
A.4. Exemple de points particuliers dans le fichier OSM. 135
A.5. Exemple d'une route dans le fichier OSM. 135
A.6. Exemple d'une zone dans le fichier OSM. 136
A.7. Fichier .osm ouvert à l'aide d'un logiciel de GIS. 136
A.8. Informations portées par les routes dans OpenStreetMap. 137
A.9. Carte avec différenciation du type des routes et filtrage. 137
A.10. Changement vers le repère local tangent. 138
A.11. Carte représentée dans le repère local tangent. 139

Licence CC BY-NC-ND 3.0

Liste des symboles

AGM Attributed Graph Matching

API Application Programming Interface

ASL Autonomous System Lab

DMA Delta Medial Axis

ECEF Earth Centered, Earth Fixed

EKF Extended Kalman Filter

ENU East North Up

ETHZ École Polytechnique Fédérale de Zurich

LIDAR Light Detection And Ranging

MAP Maximum A Posteriori

MCS Maximal Common Sub-graph

MNT Modèle Numérique de Terrain

OSM OpenStreetMap

PAVIN Plateforme Auvergne pour Véhicules INtelligents

QAP Quadratique Assignment Problem

RANSAC RANdom SAmple Consensus

RGB-D Camera couleur RGB avec un canal supplémentaire pour la profondeur

SAM Smoothing And Mapping

SGM SubGraph Matching

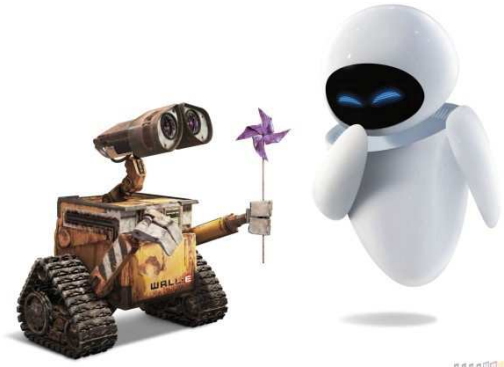
SIG Système d'Information Géographique

SLAM Simultaneous Localization And Mapping

TAM Transformée en Axe Médian

Voxels Volumetric pixels

Introduction générale



[1]

Contexte : Coopération aéro-terrestre

Les systèmes sans pilote trouvent leur utilité dans les travaux répétitifs, dépréciant ou dangereux, aussi désignés par les trois « D » (dull, durty, dangerous). Ainsi, les domaines d'applications sont variés, des robots de production, aux robots tondeuses, en passant par les robots de surveillance de la sécurité civile, et les drones militaires. Les robots sont de plus en plus performants et autonomes, cependant, un seul robot est souvent limité dans ses capacités d'action, de perception ou de déplacement. Ainsi, les systèmes multirobots sont un domaine en plein essor depuis la fin des années 1990, et encore plus de nos jours, du fait des avancées technologiques et de l'intelligence artificielle.

Considérons en particulier la robotique d'intervention qui a pour but de développer des robots utilisables lors de missions où une présence sur place est obligatoire. C'est le cas des missions de secours après sinistres et de luttes contre les incendies. Dans ces milieux, souvent inconnus et dangereux, l'intervention doit être rapide et ne pas mettre la vie des équipes de secours en danger. Il y a donc un grand intérêt à utiliser une équipe de robots et tout particulièrement des robots avec des capacités différentes. Le cas de la coopération aéro-terrestre offre notamment des possibilités intéressantes grâce à la complémentarité des agents. Les caractéristiques des robots changent grandement selon leur taille et leur conception (micro véhicule, voilure fixe,

1. « Wall-e and Eve », personnages du film d'animation Wall-e, Pixar.

voilure tournante, etc). On peut néanmoins simplifier la différence Rover/Drone [2] par les caractéristiques suivantes (Tableau 0.1) :

	Rover	Drone
Capacité de charge	++	-
Contraintes de mobilité	-	+
Capacité d'intervention	+	-
Autonomie énergétique	+	-
Champ de vision	+ -	++

Table 0.1. – Complémentarité des robots terrestre et aérien.

Pour bénéficier pleinement de la complémentarité de ces robots, il est nécessaire de spécifier des missions adaptées. Les missions sont souvent réparties de façon séquentielle et partagée :

1. Le drone réalise la cartographie globale et revient au point de départ.
2. Les données sont traitées par l'utilisateur.
3. Le robot terrestre part ensuite en mission avec une connaissance à priori.

Dans ce cas, il n'y pas de réelle coopération. Pour un gain de temps et d'autonomie, les robots doivent travailler ensemble. La modélisation de l'environnement à partir de données terrestres et aériennes est alors cruciale. Si la construction de cartes de l'environnement peut être le but de la mission des robots (cas de la robotique d'exploration), elle est surtout nécessaire à la prise de décisions. C'est cette carte de l'environnement qui permet aux robots de coopérer efficacement.

Ces dernières années, de nombreux projets de recherche ont mis en avant la coopération multirobots dans des missions d'intervention (Eurathlon, TRADR) ou de surveillance (PEA Action, R-Discover).

Dans cette étude nous nous plaçons dans le cadre suivant :

Le robot terrestre possède l'équipement nécessaire pour intervenir mais ne dispose pas à priori de la carte de l'environnement. Le drone a pour objectif de l'aider dans la cartographie de l'environnement et dans la détection d'objectifs à atteindre. Ces deux missions se font simultanément et la coopération se fait grâce à une communication entre les deux robots et sans intervention extérieure.

2. Le rover désigne généralement un robot terrestre roulant, et le drone désigne un système aérien à voilure fixe ou tournante, sans pilote.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Problématique

L'objectif de cette thèse est de fournir une perception augmentée et un référentiel commun à un système hétérogène aéro-terrestre. Nous avons, plus particulièrement, proposé une solution au problème de la cartographie coopérative pour fournir au robot terrestre une carte de son environnement afin qu'il puisse effectuer sa mission. La problématique peut être résumée ainsi :

Comment utiliser des robots terrestres et aériens pour construire une carte de navigation au sol à l'aide d'informations provenant des deux points de vue ?

Cette problématique soulève plusieurs défis liés à la variation importante des points de vue, aux différentes modalités de perception, et aux référentiels multiples.

Contribution

Pour tenter de répondre à cette problématique, nous proposons une approche en plusieurs points :

- La définition d'une représentation unifiée de la perception de l'espace navigable.
- La caractérisation d'une carte hybride modulaire adaptée aux spécifications des systèmes multirobots.
- L'élaboration de méthodes de création de cette carte à partir de données terrestres et aériennes.
- La mise en correspondance de ces cartes dans un processus hiérarchique à l'aide d'une mesure adaptée à des sources hétérogènes.
- Le développement d'une méthode de fusion de ces cartes pour la création d'une carte enrichie des informations les plus pertinentes.

Organisation du manuscrit

Le document est organisé comme suit :

- Le 1er chapitre présente un état de l'art des domaines à l'œuvre dans la cartographie coopérative aéro-terrestre.
- Le 2e chapitre décrit le modèle de représentation élaboré pour la cartographie ainsi que les méthodes proposées pour sa construction à partir des images provenant d'un drone, d'un robot terrestre, ou d'une base de données.
- Le 3e chapitre détaille les différentes étapes du processus de mise en correspondance et de la fusion des cartes.
- Le 4e chapitre présente les résultats en environnements simulés et réels. Les différents éléments de notre approche y sont évalués et discutés.

Licence CC BY-NC-ND 3.0

1. État de l'art de la cartographie multirobots

Sommaire

1.1. Localisation et cartographie simultanées 13
1.1.1. Optimisation conjointe 14
1.1.2. Pré-traitement dépendant du capteur 16
1.2. Représentation de l'environnement 18
1.2.1. Métrique : la géométrie de l'environnement 19
1.2.2. Topologique : les connexions entre lieux 22
1.2.3. Hybride : la complémentarité des deux approches 25
1.2.4. Récapitulatif 28
1.3. Coopération multirobots 29
1.3.1. Système multirobots homogène ou hétérogène 30
1.3.2. Architecture centralisée, décentralisée ou distribuée . . . 32
1.3.3. Planification externe ou autonome 34
1.4. Les défis à relever 35
Conclusion 37

Introduction

Pour qu'un robot se déplace et intervienne dans le monde réel, il a besoin de capteurs pour percevoir son environnement ainsi que d'une carte. La nécessité d'une carte a deux origines. Premièrement, une carte est souvent nécessaire à la mission pour établir un chemin et définir une destination ou un objectif. Deuxièmement, une carte permet de réduire les erreurs de localisation en utilisant des points de référence (landmarks) et en revisitant des lieux déjà explorés.

Dans certaines applications, une carte, ou un ensemble de points de références, est connu à priori. C'est le cas des voitures autonomes qui utilisent le réseau routier ou des entrepôts robotisés d'Amazon qui exploitent un plan de bâtiment. La localisation par GPS peut également être considérée comme faisant partie de ces applications, puisque le robot connait la constellation des satellites et les utilise comme points de repère. Lorsque la carte n'est pas connue à priori, elle doit être construite par le robot lui-même. Pour construire la carte, le robot doit placer les éléments observés par ses capteurs dans cette carte et pour cela, le robot doit être capable de se localiser dans cette carte en cours de construction. En effet, la carte aide à la localisation, et inversement, la localisation est nécessaire à la cartographie pour placer les nouveaux éléments. Ce problème circulaire est connu sous le nom de « Localisation et cartographie simultanées » ou SLAM pour « Simultaneous Localization And Mapping ». De part sa complexité et sa nécessité, ce problème est un domaine à part entière de la recherche en robotique. Dans la suite de ce mémoire, c'est le terme de SLAM que nous utiliserons principalement.

Le SLAM est dépendant des modalités de perception du système mais également de l'application visée et du type d'environnement. En effet, les points de repère perçus sont différents selon les capteurs utilisés : télémétrie (laser, sonar, radar), vision (caméra mono, stéréo, couleur, RGB-D), etc. Ces mesures de capteurs extéroceptifs sont souvent fusionnées avec des mesures proprioceptives telles que l'odométrie ou les centrales inertielles. Dans certaines applications spécifiques, des capteurs dédiés sont utilisés pour la détection de gaz toxique ou la cartographie thermique [Roldan et al., 2016]. Le type d'environnement affecte également le modèle de carte et la difficulté de la cartographie. Dans un environnement extérieur ouvert, l'utilisation du GPS est possible mais le nombre de points de repère est souvent limité, contrairement aux environnements urbains qui fournissent une grande quantité de repères visuels (fenêtres, portes, voitures, etc). Les environnements intérieurs, eux, sont plus structurés mais l'utilisation du GPS est impossible.

La problématique du SLAM est un assemblage complexe de sous-problèmes plus ou moins interdépendants. Les données provenant des capteurs étant imparfaites (dérives, bruits, imprécisions,...), il faut gérer les erreurs induites sur la localisation et la cartographie. Il faut pouvoir associer les points de repère perçus avec ceux de la carte globale en cours de construction, malgré les imperfections des capteurs et le manque de précision de la localisation. La cartographie robotisé est un domaine complexe qui a fait des progrès incroyables dans les trente dernières années

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

et qui affecte d'autres aspects de la robotique comme par exemple la coopération multirobots.

Dans ce chapitre, nous commencerons par présenter la structure générale du SLAM et sa forme canonique. La cartographie sous-entend la création d'un modèle de l'environnement qui peut prendre plusieurs formes : des données capteur brutes ou fondée sur des primitives géométriques. Elle peut également représenter les connections entre des lieux particuliers. Nous aborderons ensuite plus en détail ces approches et les avantages obtenus en les combinant pour former une représentation hybride. Nous traiterons également les problèmes et les intérêts liés aux systèmes multirobots. Enfin, nous proposerons une liste non exhaustive des défis qu'il resterait à relever.

1.1. Localisation et cartographie simultanées

Le SLAM (parfois appelé SAM pour Smoothing And Mapping) est un domaine étudié depuis longtemps, les premiers travaux majeurs datant des années 90 avec l'étude de R.C. Smith et P. Cheeseman [Smith and Cheeseman, 1986], et plus tard, les travaux de Leonard et Durrant-Whyte [Leonard and Durrant-Whyte, 1991]. Dans l'état de l'art récent [Cadena et al., 2016], les auteurs montrent que le sujet est toujours d'actualité bien que le problème soit considéré comme résolu dans des conditions simples. Le SLAM y est défini comme la construction simultanée d'une représentation de l'environnement et l'estimation de l'état du robot dans ce modèle. C'est une méthode incrémentale qui construit la carte globale au fur et à mesure des perceptions locales du robot, tout en gardant une estimation de la position du système dans cette carte.

De manière générale, le SLAM est constitué de deux grandes parties : un bloc avant (front-end) et un bloc arrière (back-end). Le bloc avant traite les données brutes des capteurs pour les adapter et les filtrer pour le bloc arrière, qui calcule l'optimisation conjointe de la cartographie et de la localisation.

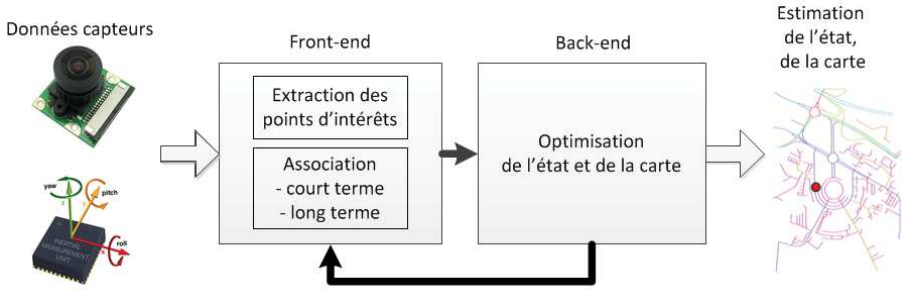


Figure 1.1. – Structure générale d'un algorithme de SLAM.

Licence CC BY-NC-ND 3.0

1.1.1. Optimisation conjointe

Considérons un robot se déplaçant dans un environnement inconnu. Ce robot est capable, grâce à ses capteurs embarqués, d'observer un certain nombre de points de repère appelés **amers**. Le schéma suivant (Figure 1.2) illustre le problème sous forme d'un graphe où les positions successives du robot sont reliées aux amers par des contraintes de déplacements et d'observations. Ce graphe est appelé « *factor graph* ».

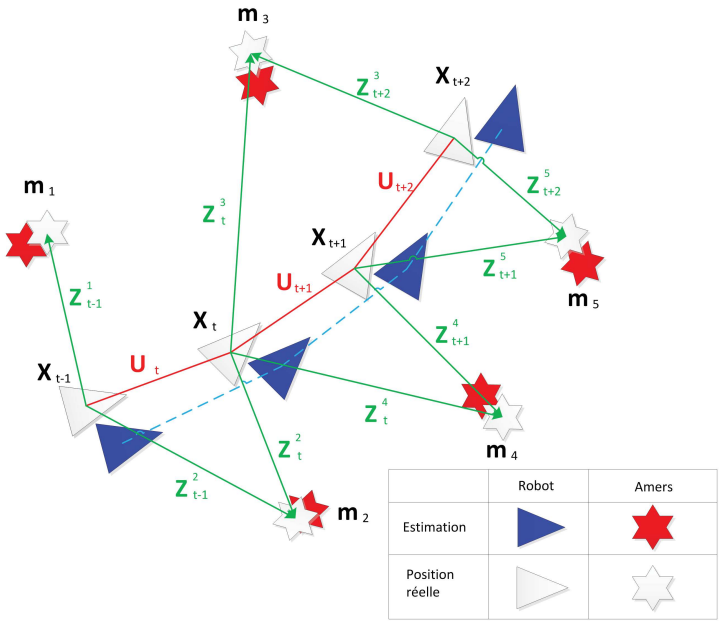


Figure 1.2. – Représentation du problème de SLAM sous forme de *factor graph*.

À chaque instant t, on peut définir :

- x_t : le vecteur d'état, il contient notamment la pose du robot
- u_t : le vecteur de contrôle, il mène le robot de l'état x_{t-1} à l'état x_t
- m_i : le vecteur contenant la position de l'amer i
- z_t : l'observation des amers à l'instant t

On peut ainsi définir les éléments suivants :

- $X_t = \{x_0, x_1, ..., x_t\}$: l'ensemble des vecteurs d'état jusqu'à l'instant t
- $U_t = \{u_0, u_1, ..., u_t\}$: l'ensemble des contrôles jusqu'à l'instant t
- $Z_t = \{z_0, z_1, ..., z_t\}$: l'ensemble des observations jusqu'à l'instant t
- $M = \{m_1, m_2, ..., m_n\}$: la carte de l'environnement contenant la liste des amers.

Les roboticiens utilisent une approche probabiliste pour remplacer la question cruciale en robotique « *Où suis-je ?* » par la question « *Où suis-je susceptible d'être dans*

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

la carte la plus probable du monde que j'ai observé jusqu'à maintenant ? ». La formulation canonique du SLAM provient certainement de l'article [Lu and Milios, 1997] bien que depuis, beaucoup d'autres travaux ont amélioré l'efficacité de la résolution du problème.

La localisation

Le problème de localisation revient à estimer la trajectoire la plus probable d'avoir été exécutée $X_t^\star$ connaissant la carte M, les observations Z_t et les contrôles U_t :

$$X_t^\star = \underset{X}{argmax}\ p(X|Z_t, U_t, M) \tag{1.1}$$

Cette formulation, appelée *Maximum A Posteriori* (MAP), utilise toutes les données de l'historique pour estimer la trajectoire. On obtient ainsi une estimation globale de la pose. La résolution de ce problème est cependant très complexe. Une formulation locale peut être établie pour simplifier l'algorithme en n'utilisant que les données à l'instant t.

$$x_t^\star = \underset{x}{argmax}\ p(x|z_t, u_t, M)$$

L'utilisation de cette méthode simplifie énormément les calculs puisqu'il y a moins de mesures à traiter. Cependant, l'incertitude de la pose augmente beaucoup plus vite à cause de l'accumulation des incertitudes locales.

La cartographie

Le problème de cartographie consiste à mettre à jour la carte M de l'environnement avec de nouveaux amers, connaissant la pose du robot et les observations des capteurs :

$$M = \underset{m}{argmax}\ p(m_i|Z_t, X_t) \tag{1.2}$$

Une localisation précise et une bonne connaissance des modèles des capteurs utilisés sont essentielles pour positionner correctement les amers dans la carte.

Le SLAM

Comme son nom l'indique, le SLAM est la résolution conjointe des deux problèmes d'optimisation précédents. Ainsi, à chaque instant t, l'algorithme doit résoudre le problème d'optimisation suivant :

$$(X_t^\star, M^\star) = \underset{(X,m)}{argmax}\ p(X, M|Z_t, U_t, x_0) \tag{1.3}$$

La solution est la trajectoire la plus probable dans la carte la plus probable construite à partir de toutes les perceptions et du point de départ.

Licence CC BY-NC-ND 3.0

Souvent, la trajectoire du robot et l'ensemble des amers sont contenus dans la même variable $\mathcal{X} = \{X, M\}$. En utilisant le théorème de Bayes et en faisant l'hypothèse que les observations sont indépendantes, on peut reformuler l'équation 1.3 ainsi :

$$\mathcal{X}^\star = \underset{\mathcal{X}}{argmax}\ p(Z|\mathcal{X})p(\mathcal{X}) = \underset{\mathcal{X}}{argmax}\ p(\mathcal{X}) \prod p(z_k|\mathcal{X}_k) \tag{1.4}$$

Ce problème peut être résolu de plusieurs façons. Il y a par exemple des approches par filtrage, comme le filtrage particulaire [Grisetti et al., 2007] ou le filtrage de Kalman étendu (EKF). Le filtre à plusieurs états proposé par Mourikis et Roumeliotis [Mourikis and Roumeliotis, 2007] est particulièrement efficace car il fusionne la vision et les informations inertielles et utilise une fenêtre d'observation sur un horizon temporel étendu. Les solveurs utilisant des linéarisations successives sont également très répandus. Certains utilisent la faible densité de la matrice induite par la topologie du graphe de facteurs pour définir une approche incrémentale, même pour l'optimisation globale [Kaess et al., 2008].

1.1.2. Pré-traitement dépendant du capteur

Dans la plupart des applications en robotique, il est difficile, voire impossible, d'écrire la mesure du capteur comme étant une fonction analytique de l'état $z_k = h_k(\mathcal{X}_k)$. Par exemple, dans le cas d'une caméra, il est délicat d'exprimer l'intensité de chaque pixel d'une image en fonction de l'état du robot et de la carte, et il en va de même avec des capteurs plus simples comme les lidars (*light detection and ranging*).

C'est pour cela que la plupart des algorithmes de SLAM utilise un bloc avant qui traite les points de repère (aussi appelés points saillants ou *features* en anglais) à partir des données provenant des capteurs. Pour un SLAM utilisant une caméra, le bloc avant va extraire dans les images des points ou des régions d'intérêt. derniers seront plus faciles à traiter pour le bloc arrière.

Le bloc avant est aussi en charge d'associer les observations aux points de repère. En effet, un même point de repère peut être vu depuis plusieurs états. Par exemple, dans la figure représentant le *factor graph* (Figure 1.2), les observations Z_t^2 et Z_{t+1}^2 ont été associées au même amer m_2 à deux instants différents. Cette association est très importante. À court terme, elle permet de suivre les amers et ainsi générer les mesures nécessaires à l'algorithme. On peut alors faire une triangulation des points de repère détectés dans une image pour estimer leur position 3D dans le monde. Un manque d'associations ou des erreurs d'associations des points de repère induisent des erreurs dans l'estimation de la position et donc également dans la carte. Heureusement, plusieurs méthodes permettent de réduire l'impact de ces mauvaises associations, entre autre, l'algorithme RANSAC (*RANdom SAmple Consensus*) permet de rejeter les points qui ne satisfont pas l'hypothèse que le plus grand nombre de points satisfait. C'est un filtrage statistique très souvent utilisé.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

L'association des amers est également utilisée à long terme. Le but est alors de reconnaitre un lieu par lequel le robot est déjà passé. Ce procédé est appelé fermeture de boucle (*loop closure*). Cela permet de corriger la dérive de l'algorithme en réajustant la carte et la position du robot comme montré dans la figure 1.3, extraite de l'algorithme ORB-SLAM [Mur-Artal et al., 2015]. La carte est représentée par un nuage de points en noir, la trajectoire estimée du robot est en vert et les observations à cet instant sont en rouge. L'image du haut représente la carte à l'instant de la détection de la fermeture de boucle. Cette fermeture de boucle, représentée par un trait bleu, permet de détecter une incohérence dans la position du robot. Cette dernière est donc modifiée et cette modification est propagée aux observations précédentes pour corriger la carte. La carte corrigée est représentée dans l'image du bas.

Figure 1.3. – (haut) Détection de la fermeture de boucle en bleu (bas) Carte corrigée. La carte est représentée par un nuage de points en noir. La trajectoire estimée du robot est en vert et les observations à cet instant sont en rouge [Mur-Artal et al., 2015].

Cette capacité à reconnaitre les lieux visités est un point important des algorithmes de SLAM modernes. Plusieurs travaux se concentrent sur cette problématique, le défi étant d'arriver à détecter des lieux identiques, quelles que soient les conditions d'observation. Pour faire face aux problèmes de variations de luminosité et d'éclairage (entre jour et nuit par exemple), plusieurs approches ont été proposées. La détection

Licence CC BY-NC-ND 3.0

peut se faire sur une séquence d'images successives [Milford and Wyeth, 2012] ou en utilisant une bibliothèque d'images obtenues lors d'expériences tout au long de l'année [Churchill and Newman, 2013]. Cette dernière approche permet de gérer les changements de saison et l'apparition de neige dans les images. La détection de fermeture de boucle peut également être utilisée pour les capteurs lasers à l'aide de descripteurs [Tipaldi and Arras, 2010]. Les deux approches métriques et visuelles peuvent être fusionnées pour une meilleure détection [Ho and Newman, 2006]. La difficulté de la fermeture de boucle peut également venir des éléments dynamiques de l'environnement. En effet, un espace peut avoir changé entre deux visites. S'il y a des éléments dynamiques comme des piétons ou des voitures, il est fort probable qu'ils changent de place ou quittent ce lieu. Pour résoudre ce problème, il est possible de détecter les éléments mobiles et de ne pas les ajouter à la carte statique [Wang et al., 2007]. Un état de l'art complet à propos de la reconnaissance de lieux est proposé dans l'article [Lowry et al., 2016].

Ce processus de pré-traitement des informations dépend des capteurs utilisés et influence la représentation de l'environnement.

1.2. Représentation de l'environnement

Le choix du type de représentation ne modifie pas la structure générale utilisée pour le SLAM. Ce choix se fait au regard de plusieurs éléments :

— Les modalités de perception (capteurs, capacité de détection, ...).
— La mobilité du robot (terrestre, aérien, ...).
— Le type d'environnement (intérieur, extérieur, structuré, ...).
— L'objectif de la cartographie (navigation, rendu visuel, ...).

La mobilité d'un système peut permettre de choisir la dimension de la représentation. Ainsi un robot qui se déplace sur le sol peut se satisfaire d'une carte 2D, comme une carte routière, puisqu'il n'évolue pas dans la 3$^{\text{éme}}$dimension. Cependant, si les informations disponibles dans ce plan ne sont pas suffisantes, il peut être préférable de représenter le monde en 3D pour accéder à des amers plus nombreux et plus distinctifs. Le SLAM utilisé est ainsi plus robuste. La représentation en 2D est plus simple et est étudiée depuis assez longtemps pour avoir permis la définition d'une norme dédiée à la robotique en 2015 par *IEEE Robotics and Automation Society* [IEEE, 2015]. Cette norme spécifie les modèles de données cartographiques et les formats de sauvegarde. Le document fournit des spécifications pour les représentations 2D métriques et topologiques mais les définitions peuvent être étendues en 3D. Nous nous appuierons sur ce document pour présenter les 3 grands types de représentations : *métrique*, *topologique* et *hybride*. Ces représentations seront par la suite appelées *cartes.*

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

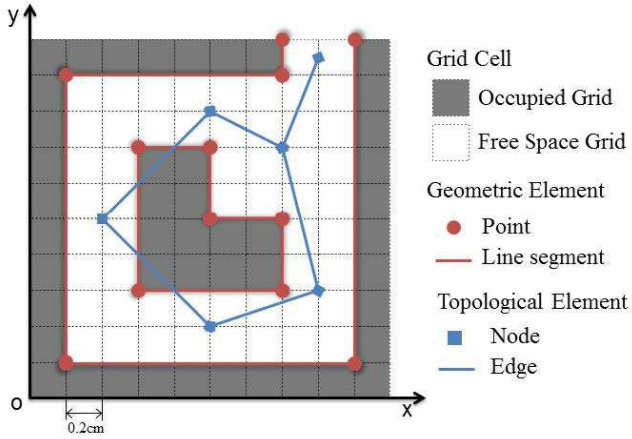


Figure 1.4. – Les 3 grands formats de représentation 2D [IEEE, 2015].

1.2.1. Métrique : la géométrie de l'environnement

Les cartes dites métriques sont des représentations qui ont pour but de modéliser la géométrie de l'environnement et, plus précisément, la géométrie et la position des obstacles et des frontières.

Une carte métrique est une collection d'éléments cartographiques (par exemple, des cellules, des points et des segments de ligne) avec la propriété suivante : étant donné deux éléments, a et b dans la carte, et la définition d'une distance métrique, la distance entre a et b peut toujours être calculée.

Il convient de noter que le terme « toujours » précédent est ce qui rend les cartes métriques différentes des cartes topologiques, où la distance métrique ne peut être calculée que pour certains éléments.

Ces cartes métriques se divisent en 2 familles : les représentations continues et les représentations discrètes.

Les cartes continues

Ces cartes sont constituées d'une liste d'éléments définis de manière continue (position d'un point, d'un segment). Elles sont aussi appelées « **carte géométrique** ». Un exemple est présenté en rouge dans la figure 1.4. On peut distinguer deux types de représentations continues selon les éléments stockés : des données brutes ou des primitives géométriques possédant des propriétés particulières (descripteurs).

$$M = \{p_i\},\ p_i = [position_i, propriétés_i]$$

Ces cartes sont majoritairement utilisées dans les algorithmes de SLAM visuel où les amers sont des points d'intérêt détectés dans une image, c'est le cas de l'algorithme ORB-SLAM [Mur-Artal et al., 2015]. Les primitives géométriques peuvent

Licence CC BY-NC-ND 3.0

également être des lignes comme proposé dans [Vidal et al., 2011] et présenté dans l'image de droite (Figure 1.5). Les cartes continues sont également appropriées pour les capteurs de type lidar car elles sont adaptées aux données brutes sous forme de points renvoyés par ces capteurs, présentées dans l'image de gauche (Figure 1.5).

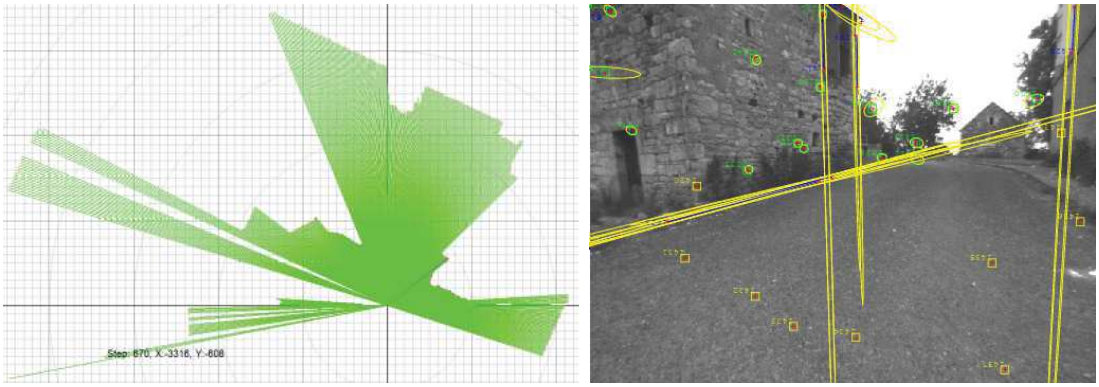


Figure 1.5. – Cartes géométriques : (gauche) Lidar 2D (droite) Points et lignes 3D.

Cette représentation a l'avantage de ne pas être limitée dans la précision des mesures. Cependant, le nombre d'informations à stocker augmente rapidement avec la création de la carte et l'espace mémoire peut devenir une limitation.

Les cartes discrètes

Ces cartes représentent l'environnement comme un assemblage de cellules accolées pour former une grille.

Si l'information cartographiée est la traversabilité du terrain, ces cartes sont aussi appelées « **grilles d'occupation** », un exemple est présenté dans la figure 1.4. Les valeurs stockées dans les cellules de la grille correspondent alors à la probabilité d'occupation (0 = occupé, 1 = libre, 0.5 = inconnu).

Les cartes discrètes ne se limitent pas seulement à la traversabilité et peuvent être utilisées pour représenter la hauteur du sol, on parle alors de **carte d'élévation** ou de **Modèle Numérique de Terrain** (MNT). Il s'agit d'une représentation 2D d'un environnement 3D, qu'on définit parfois comme des cartes 2.5D. La figure suivante (Figure 1.6) montre les trois types de cartes discrètes [Drouilly, 2015] [1].

En 3D, les cartes discrètes prennent la forme de grilles 3D dont les cellules sont des cubes appelées *voxels*. Stocker toutes ces informations 3D prend énormément de place mémoire même s'il existe des représentations compactes compressées.

Cette représentation discrète permet d'effectuer rapidement (en 2D) des calculs tels que le filtrage ou la planification de trajectoire. Elle est également plus facile à interpréter pour l'utilisateur car elle est dense contrairement à certaines cartes continues.

1. Je remercie Romain Drouilly pour cette image explicite présente dans sa thèse [Drouilly, 2015]

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

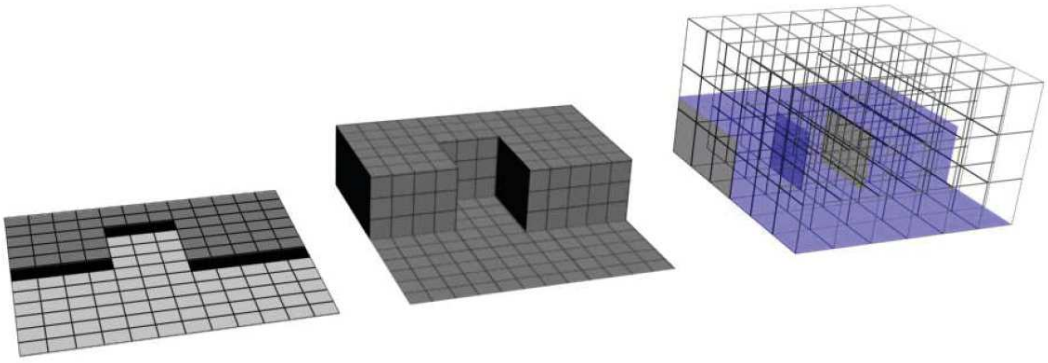


Figure 1.6. – Cartes discrètes : a) Grille 2D, b) MNT 2.5D, c)Voxels 3D.
La couleur de chaque cellule indique la probabilité de la présence d'obstacle.

Discussion

Les représentations métriques continues peuvent s'adapter à tout type d'environnements car elles sont construites sur des primitives géométriques sans volume et ne soufrent pas de la discrétisation de la représentation sous forme de grille. Cependant, cet avantage les rend difficiles à utiliser pour la planification de trajectoire. Il y a beaucoup d'espaces vides et il est parfois complexe de déterminer les espaces navigables. Cette représentation est exigeante en terme de mémoire puisque toutes les perceptions du robot sont stockées. Elle peut être difficile à lire pour l'utilisateur si elle n'est pas assez dense et ne fait pas apparaitre clairement les obstacles. C'est pourquoi cette représentation est plus adaptée aux capteurs fournissant un grand nombre d'informations telles que les caméras et les télémètres laser.

Les représentations métriques discrètes permettent de représenter l'espace de manière dense, avec moins de mesures et intègrent une estimation statistique de la confiance dans les mesures. De plus, contrairement aux représentations continues constituées de données brutes, elles fournissent des informations sur les probabilités d'occupation d'un espace, et indiquent la position des obstacles et des zones non observées. Cela permet de définir assez facilement l'espace navigable et de calculer des trajectoires. La représentation sous forme de grille, avec la seule information d'occupation, permet une mise à jour rapide pour gérer les éléments dynamiques, ainsi qu'une fusion immédiate de divers types de capteurs. Certain type de ces représentations cherchent à définir des structures de stockage optimisée qui permettent de réduire le cout en mémoire, en regroupant les cellules de même valeur : le *Quadtree* (en 2D) l'*Octree* (en 3D). Cependant, ce type de stockage rend la création incrémentale plus complexe. La taille de la discrétisation étant définie à l'avance, la carte n'est pas capable de s'adapter à la taille des obstacles, et donc son efficacité diminue dans de grands espaces vides. Les représentations métriques discrètes sont également moins appropriées pour la résolution du problème de SLAM en particulier lorsqu'il y a des boucles. La structure probabiliste à l'échelle de la cellule permet de facilement intégrer les mesures de capteurs bruitées mais, lors de la résolution du SLAM, il n'existe pas de lien direct avec l'incertitude liée à la position du robot. Cela pose

Licence CC BY-NC-ND 3.0

problème lors de la fermeture de boucle. En effet, il n'existe pas, à l'heure actuelle, d'algorithme temps réel de rétro-propagation des corrections d'erreurs sur la grille. Lors de l'optimisation globale, toute la carte doit être recalculée et ce processus prend beaucoup de temps. Une piste envisagée est d'utiliser une représentation hybride, un graphe de grilles d'occupations locales où l'optimisation globale est faite sur le graphe topologique. La carte globale est alors obtenue grâce à une fusion des cartes locales [Ratter and Sammut, 2015][Schmuck et al., 2016].

1.2.2. Topologique : les connexions entre lieux

Une carte topologique est une représentation sous forme de graphes d'un environnement, comprenant des nœuds N et des arêtes E, et leurs relations de connectivité. Les nœuds d'une carte topologique représentent des mesures caractéristiques de l'environnement (par exemple, des mesures de capteurs, des lieux distinctifs, des cartes métriques, etc.). Les arêtes d'une carte topologique représentent des connexions directes entre deux nœuds voisins qui la définissent. Un exemple de carte topologique est présenté en bleu dans l'image de présentation (Figure 1.4) et dans la figure 1.7.

$$M = \{N, E\},\ N = \{n_i\},\ E = \{(n_i, n_j)\},\ (i, j) \in N^2$$

Un exemple quotidien d'une carte topologique est celui couramment utilisé pour les réseaux de métro (ou de bus). Les informations primaires sont des nœuds dans la carte (stations) et la connectivité entre eux (lignes de métro (ou de bus) entre les stations). Dans ce type de cartes, les positions relatives et les distances entre les éléments sont moins importantes que leurs connexions. Un exemple lié à la robotique est une carte d'un intérieur de bâtiment, où les nœuds de la carte indiquent des salles, et les arêtes représentent des itinéraires possibles pour naviguer d'une pièce à l'autre, comme représenté dans la figure 1.7.

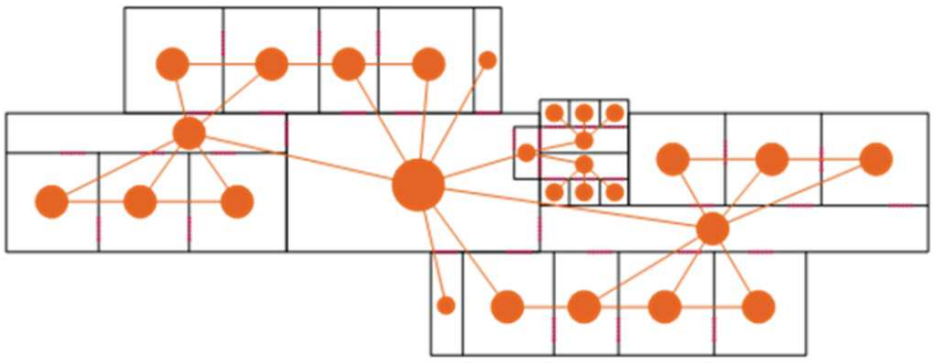


Figure 1.7. – Carte topologique d'un bâtiment.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Les cartes topologiques sont apparues avec les travaux de Kuipers qui a fait le choix de représenter l'environnement grâce à un graphe de lieux et utilise les arêtes pour définir les mouvements possibles d'un robot [Kuipers, 1978]. Les cartes topologiques sont simplement une liste d'éléments remarquables et de leur connexions, mais il n'existe pas de définition précise de ce que sont ces éléments. Les significations diffèrent selon les approches. Plusieurs critères peuvent servir à différencier les cartes topologiques :

— La nature des nœuds (les lieux) et la façon de les ajouter à la carte.
— La nature des arêtes.
— Le type et la quantité des informations portées par le graphe.

Notion de lieux et de connexions

La reconnaissance de lieux ou de points de référence constitue la capacité principale de la navigation. Chez l'Homme et chez les animaux, la représentation mentale de l'environnement (aussi appelée « cartes cognitives ») est construite comme un ensemble de lieux et de points de repère distincts auxquels sont associées des relations d'adjacences et de distances. Les cartes topologiques sont ainsi proches de la représentation mentale de l'environnement qu'utilisent les humains. Les roboticiens ont cherché à définir des lieux distinctifs qui pourraient être reconnus sans ambigüité afin de localiser le robot.

Les lieux peuvent être globalement définis de deux manières différentes : à partir des mouvements du robot (sa pose) ou de ses perceptions.

Dans le premier cas, les nœuds correspondent à un échantillonnage des différentes poses du robot et les arêtes sont alors les mouvements entre deux poses. Cette représentation est appelée « *pose graph* ». Les nœuds de ce type de cartes sont souvent enrichis des informations métriques perçues à cette pose. Pour éviter de surcharger la carte, un critère d'échantillonnage pour la création des nœuds est souvent mis en place. Il peut être défini à partir d'un déplacement minimal entre deux nœuds ou lorsque les deux perceptions sont suffisamment différentes [Mur-Artal et al., 2015] [Chapoulie et al., 2012] [Korrapati and Mezouar, 2014]. Un exemple est donné dans la figure présentant différentes cartes topologiques (Figure 1.8), la succession des flèches représente les poses successives du robot et leur relation.

Dans le deuxième cas, le robot peut associer un nœud à chaque point de repère perçu. Les connexions peuvent alors représenter différentes notions. Elle peuvent définir la connexité [Drouilly et al., 2015], ou l'inter-visibilité. Les perceptions du robot peuvent également être utilisées pour détecter des lieux distinctifs comme des intersections, des impasses ou des salles [Beeson et al., 2005], ou des signatures visuelles [Labbani-Igbida et al., 2011][Bazeille and Filliat, 2011]. Dans la figure 1.8, le graphe en rouge a ses nœuds positionnés aux coins et aux intersections et les arêtes définissent des adjacences directes pratiques pour la navigation. Dans le graphe bleu, les nœuds correspondent aux salles et aux points d'entrées, et les arêtes représentent également la connectivité entre ces points mais elles ne définissent

Licence CC BY-NC-ND 3.0

pas de chemin. Une partie des représentations topologiques utilise le squelette de l'environnement pour définir des lieux canoniques et leurs relations d'adjacences [Choset and Nagatani, 2001], cette partie sera détaillée dans le chapitre suivant.

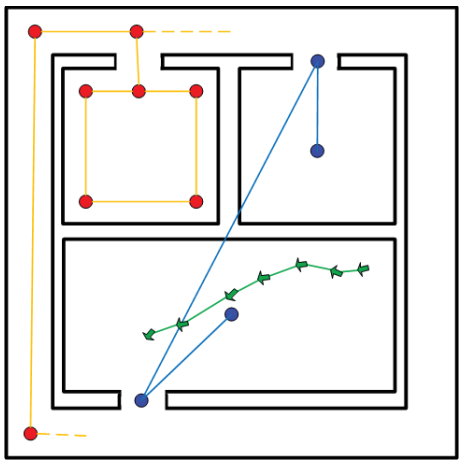


Figure 1.8. – Différents types de représentations topologiques. Le graphe en vert montre un exemple de connexion entre les poses du robot. En rouge et bleu, ce sont des représentations de l'espace mais avec des définitions différentes de leurs éléments.

Les arêtes

Les arêtes topologiques représentent toutes des notions d'adjacences mais ne sont pas définies de la même manière. Elles peuvent être bidirectionnelles ou orientées (comme les sens uniques des réseaux routiers). Elles peuvent exprimer la notion de visibilité ou d'accessibilité. Certaines ont une réalité physique (métrique) comme les routes ou les couloirs, alors que d'autres sont abstraites. De la même manière que les nœuds, les arêtes peuvent également porter des informations métriques comme la longueur du chemin ou sa largeur. Dans le cas des cartes topologiques pures, ces informations ne sont pas définies dans un repère global, elles servent plutôt de signature de l'arête, ou de poids pour la recherche du chemin le plus court. Les arêtes sont définies par les deux nœuds qu'elles connectent. Lorsqu'on veut relier plus de deux nœuds avec un seul lien, on parle d'**hyper-arête** et d'**hyper-graphe** [Pfingsthorn and Birk, 2014]. C'est le cas des réseaux de métro où une ligne (hyper-arête) dessert plusieurs stations (nœuds).

Discussion

Les définitions variables des nœuds et des arêtes offrent une grande liberté dans la représentation de l'environnement. Quelle que soit leur définition, les cartes topologiques possèdent globalement les mêmes avantages et inconvénients.

L'avantage principal des cartes topologiques est leur niveau d'abstraction élevé qui permet de s'extraire des problèmes liés aux déplacements du robot. Le robot se

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

déplace de lieu en lieu et se localise dans des référentiels locaux sans forcément utiliser de référentiel global. De plus, cette représentation facilite la planification de chemins car l'espace de recherche est grandement diminué comparé à l'espace continu. Les cartes topologiques sont proches de la représentation humaine du monde et on peut facilement y ajouter des informations sémantiques pour créer des consignes telles que « Va dans la cuisine » plutôt que « Va en (x,y) » [Beeson et al., 2010].

Cette représentation est également compacte car elle n'enregistre que les informations et les lieux importants, contrairement aux grilles d'occupation. Cet avantage est à modérer selon la définition des nœuds et des arêtes. La création d'un grand nombre de nœuds portant plusieurs informations métriques peut rendre la représentation moins économique. En suivant cette idée, des chercheurs se sont intéressés au nombre minimal de nœuds ainsi qu'à leur définition afin d'obtenir une représentation compacte de l'environnement [Beeson et al., 2005].

Cet avantage est également l'inconvénient principal de la représentation purement topologique. Le manque d'informations géométriques peut créer des difficultés lors de la planification de trajectoire ou pour la reconnaissance de lieux à cause d'un fort risque d'ambigüité perceptuelle (*perceptual aliasing*). En effet, sans informations topométriques (coordonnées cartésiennes associées aux sommets du graphe), les nœuds sont difficiles à distinguer et un processus de vérification complexe doit être mis en place lors de la création d'un nœud. Le nœud à créer est comparé à tous les autres pour savoir s'il n'existe pas déjà dans la carte et il est nécessaire de mettre à jour les relations d'adjacences.

Cette représentation est adaptée aux environnements fermés et structurés où le robot doit réaliser une navigation. Dans un espace ouvert, la topologie perd de son intérêt et limite la représentation. La cartographie topologique nécessite la plupart du temps de passer par un lieu distinctif pour y créer un nœud, contrairement aux représentations métriques qui ne sont limitées que par leurs perceptions.

Pour pallier ce problème, les deux représentations précédentes peuvent être combinées pour créer une carte hybride, aussi appelée carte combinatoire, ayant les avantages de chacune.

1.2.3. Hybride : la complémentarité des deux approches

Les représentations précédentes (carte métrique et carte topologique) ne sont pas idéales car aucune ne fournit toutes les clés pour une cartographie fiable, robuste, facile d'utilisation et adaptée à tous les environnements et capteurs. Néanmoins, chacune apporte un avantage particulier. Le tableau suivant présente globalement les avantages et les inconvénients de chaque représentation et il a l'intérêt de montrer clairement leur complémentarité.

Les cartes métriques représentent avec précision la géométrie de l'environnement mais avec une importante consommation de ressources. La représentation topologique est compacte et adaptée à la planification de trajectoires mais souffre de son

Licence CC BY-NC-ND 3.0

Critère / Type de carte	Métrique	Topologique
Taille mémoire	-	++
Environnements larges	-	++
Souplesse de cartographie	++	+
Exploitation par le robot	+	++
Visualisation par l'Homme	+	-

Table 1.1. – Complémentarité des représentations

manque d'information. La construction de représentations hybrides est née pour profiter des avantages de chacune de ces représentations. Les informations de natures différentes sont organisées sur plusieurs niveaux, ainsi les cartes hybrides sont aussi appelées cartes hiérarchiques. Il existe plusieurs manières d'associer ces deux représentations.

Cartes superposées

Les deux représentations peuvent coexister en parallèle sans relation directe, comme illustré dans la figure 1.9. Elles sont simplement superposées et utilisées indépendamment. Cette structure nécessite la construction et la mise à jour simultanée des deux représentations, ce qui demande un grand nombre de ressources. Thrun et al. [Thrun and Bucken, 1996] ont proposé de segmenter la carte métrique en régions et de construire une carte topologique de ces régions. Cette représentation nécessite, au préalable, la carte métrique globale pour construire la carte topologique. Un exemple avec une carte en 3D est présenté dans [Blochliger et al., 2018]. La carte topologique est alors un pré-traitement de la carte métrique facilitant le calcul de trajectoires.

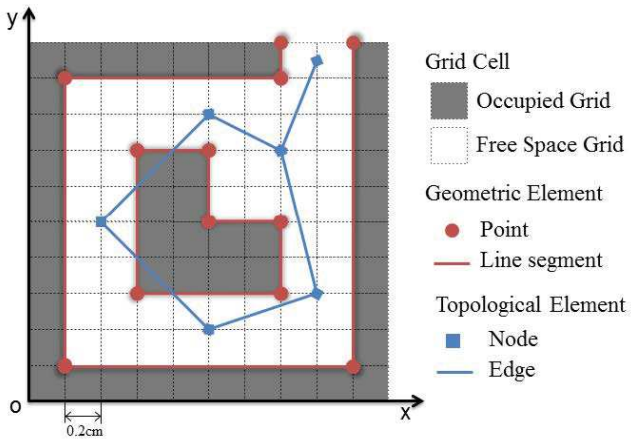


Figure 1.9. – Superposition de représentation métriques et topologiques.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Assemblage de cartes locales

Une carte peut également être créée à partir d'un ensemble de cartes métriques locales reliées grâce à un graphe topologique global. Le but est de profiter de la topologie globale pour planifier des trajectoires à l'échelle de la carte globale et d'utiliser les cartes métriques locales pour la navigation fine. Cette représentation permet de bénéficier de la précision des cartes métriques avec une forte réduction du temps de calcul nécessaire aux fermetures de boucles. Un bon exemple est la carte construite par Modayil et Kuipers dans [Kuipers, 2004], qui est présentée dans la figure 1.9. Les cartes locales sont des grilles d'occupation de lieux remarquables (intersections), qui sont positionnés aux nœuds du graphe, qui représentent ainsi la topologie de l'environnement.

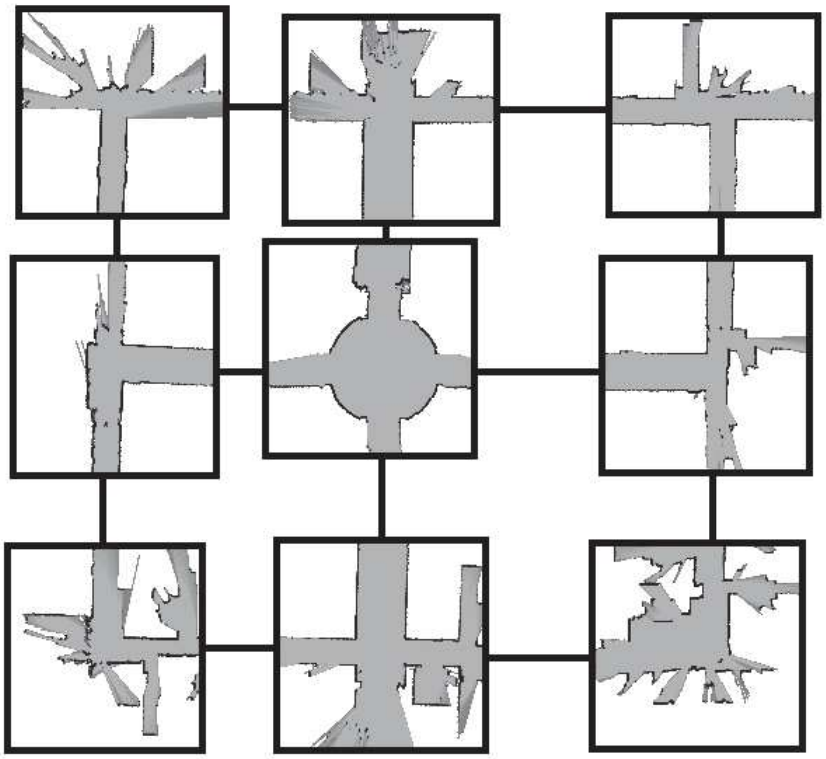


Figure 1.10. – Assemblage de cartes locales. Chaque carré noir est une carte locale obtenue grâce aux perceptions du robot. Elles sont reliées entre elles de manière à montrer les connexions entre ces lieux.

Ce type de représentation est de plus en plus utilisé car il permet de hiérarchiser les informations et les décisions. Kuipers propose en 1991 la première approche de ce type [Kuipers, 1991]. Depuis, plein d'autres ont suivi : [Lisien et al., 2005] propose un atlas hiérarchique, [Beeson et al., 2010] ajoute de la sémantique à la carte hybride et [Stephen Tully et al., 2012] propose un environnement unifié pour le SLAM hybride.

C'est ce type de représentation qui est utilisé dans cette thèse car les informations

Licence CC BY-NC-ND 3.0

topologique et métrique sont intrinsèquement reliées, ce qui permet la reconnaissance des nœuds. La représentation hybride que nous proposons sera présentée en détails dans le chapitre suivant.

1.2.4. Récapitulatif

La représentation de l'environnement pour la cartographie et la navigation est un domaine de recherche spécifique depuis le début de la robotique. Plusieurs modèles ont été proposés avec un niveau d'abstraction plus ou moins élevé. Le schéma (Figure 1.11) présente globalement la hiérarchie des représentations. Les cartes topologiques pures nécessitent un niveau de compréhension élevé de l'environnement et ne se suffisent pas à elles-mêmes. Elles doivent se reposer sur des algorithmes complexes de reconnaissance de lieux et de navigation locale. Les cartes métriques utilisent les données brutes des capteurs ou un traitement pour extraire des primitives géométriques. Elles fournissent une représentation précise de l'environnement qui nécessite une grande taille mémoire et ne sont pas toujours adaptées pour la planification de chemin. Les cartes hybrides procurent la topologie de l'environnement pour une planification haut niveau ainsi qu'une représentation métrique locale permettant une navigation locale et une identification des lieux distinctifs.

Il existe une dernière façon de représenter l'environnement à un plus haut niveau d'abstraction : l'approche sémantique. Elle doit être associée à une autre représentation (métrique ou topologique) et peut donc être classée dans les cartes hybrides. Les cartes sémantiques permettent de donner un sens et des capacités ou des caractéristiques aux éléments présents dans l'environnement [Atanasov et al., 2018]. Nous l'aborderons partiellement tout au long de notre étude.

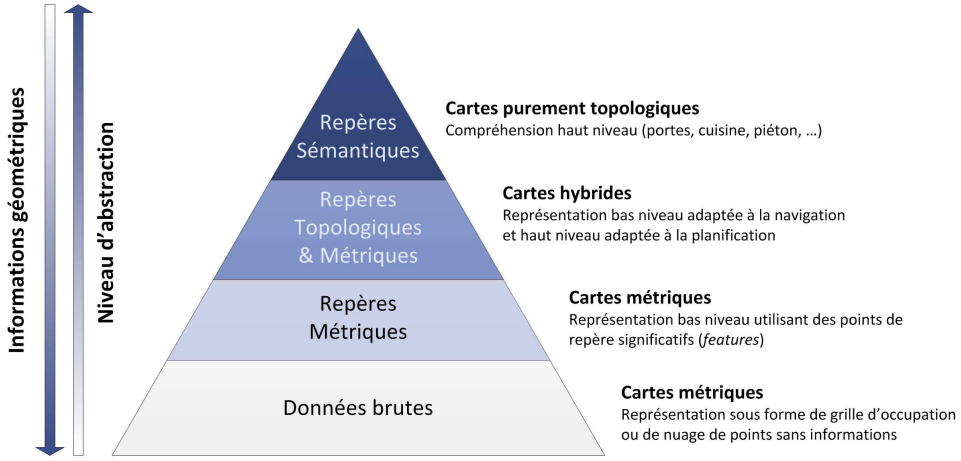


Figure 1.11. – Hiérarchie des représentations, traduit et adapté de [Adriana Tapus, 2005].

À l'heure actuelle, les cartes hybrides ne sont pas des représentations parfaites. La localisation du robot ou de certains éléments est faite dans des cartes locales avec

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

peu de garantie sur la position globale. Les cartes locales sont toujours sensibles aux erreurs de perception et d'odométrie. Pour pouvoir détecter les fermetures de boucles, il faut disposer d'un moyen pour comparer les cartes locales entre elles.

La cartographie est faite par déplacement du robot. Dans le cadre de la robotique terrestre, les robots ont des vitesses de déplacement lentes et un champ de vision assez limité. L'utilisation de plusieurs robots permet d'accélérer le processus de cartographie.

1.3. Coopération multirobots

Si les capacités individuelles d'un robot ne suffisent pas, par manque de temps, de capacité de perception ou d'action, des capacités conjointes ou complémentaires peuvent permettre d'accomplir la mission. Dans cette section, nous allons faire un tour d'horizon de différents systèmes multirobots existants, en abordant leurs avantages et leurs inconvénients. En effet, les systèmes multirobots forment un domaine à part entière dans la recherche en robotique. Ce domaine aborde un grand nombre de sujets différents : la planification de tâches, la gestion de la communication, l'allocation des ressources, l'utilisation d'informations provenant d'autres agents, etc.

L. Parker [Parker, 2003] explique que les motivations fondamentales de la recherche en systèmes multirobots sont :

1. La capacité de résoudre des problèmes qui sont intrinsèquement distribués dans l'espace, dans le temps ou dans la fonctionnalité.
2. La capacité de résoudre des problèmes plus rapidement grâce au parallélisme.
3. La capacité d'augmenter la robustesse des solutions grâce à la redondance.

Dans [Parker, 2008], il est proposé une méthode permettant de distinguer les systèmes multirobots selon les interactions entre individus (entités), voir table 1.2 :

— **Collectifs** : Les entités ont un but commun mais n'ont pas conscience des autres. Leurs actions peuvent cependant affecter celles des autres. C'est le cas des équipes en essaim (*swarm*) avec un grand nombre d'entités ou des déplacements en formation [James D, 2005].
— **Coordonnés** : Les entités n'ont pas de but commun mais, pour éviter les interférences, ils communiquent pour organiser leurs tâches. C'est le cas lorsque, par exemple, il y a moins de postes de rechargement que de robots. Une régulation des charges doit être mise en place, et les trajectoires sont calculées pour éviter les collisions [Kloder and Hutchinson, 2006].
— **Collaboratifs** : Les entités ont des buts différents mais l'interaction aide les autres agents à accomplir leur but. C'est le cas lorsque les robots partagent leurs perceptions pour mieux se localiser sans pour autant partager un but global [Parker and Fang Tang, 2006].

Licence CC BY-NC-ND 3.0

— **Coopératifs** : Les entités ont un but commun et interagissent entre elles pour accomplir ce but plus efficacement. C'est le cas lorsque plusieurs robots cartographient un environnement en se partageant la zone à quadriller [Stroupe et al., 2006] ou que plusieurs robots se regroupent pour déplacer un objet [Gerkey and Mataric, 2002].

Type d'interaction	Connaissance des autres	Objectif identique	Actions inter-dépendantes
Collective	non	oui	oui
Coordonnée	oui	non	non
Collaborative	oui	non	oui
Coopérative	oui	oui	oui

Table 1.2. – Les interactions multirobots [Parker, 2008].

Au-delà du type d'interaction, les systèmes multirobots peuvent être différenciés grâce à plusieurs critères :

— Les robots utilisés : équipe homogène ou hétérogène.
— L'architecture : centralisé, décentralisé ou distribué.
— La planification des tâches : inter-agent ou centralisée.

1.3.1. Système multirobots homogène ou hétérogène

Les premiers travaux sur les systèmes multirobots se sont penchés sur les équipes homogènes dans lesquelles les robots sont identiques. Leurs capacités de déplacement, de perception et d'action sont les mêmes. Cette homogénéité facilite la coopération dans le sens où les informations sont du même type et les points de vue sont similaires. Les jeux en équipe, et en particulier le football, ont servi de support aux recherches sur la coopération des équipes homogènes. Ainsi, on a pu voir la création de la première coupe du monde de football pour robots (*RoboCup*) en 1997 (Figure 1.12). Dans [Veloso and Stone, 1998], les vainqueurs de cette première compétition présentent les difficultés liées à cette compétition du point de vue des systèmes multirobots ainsi que les solutions qu'ils ont utilisées.

Lorsque le nombre de robots augmente, on parle de robotique en essaim. Ces équipes avec un grand nombre de robots font émerger des phénomènes de groupe. Avec un grand nombre de robots, le pilotage de chaque individu n'est pas toujours possible à partir d'une station fixe, il faut donc utiliser des entités autonomes. Le défi est alors de garder une cohérence de groupe entre des robots indépendants en utilisant

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Figure 1.12. – Photo de la Robocup (Small Size League).

des contraintes locales de communication ou de visibilité. Ces robots étant en grand nombre, ils sont souvent peu équipés ou utilisent des capteurs peu chers pour réduire les couts, ce qui rend la tâche encore plus complexe. Une autre utilisation proposée par Tuci [Tuci et al., 2006] consiste à appliquer l'expression « l'union fait la force ». Les robots développés dans cette étude sont capables de s'accrocher entre eux pour former des structures complexes leur permettant ainsi de franchir des obstacles. La figure 1.13 est une image extraite de [Mathews et al., 2012] montrant l'assemblage de robots homogènes.

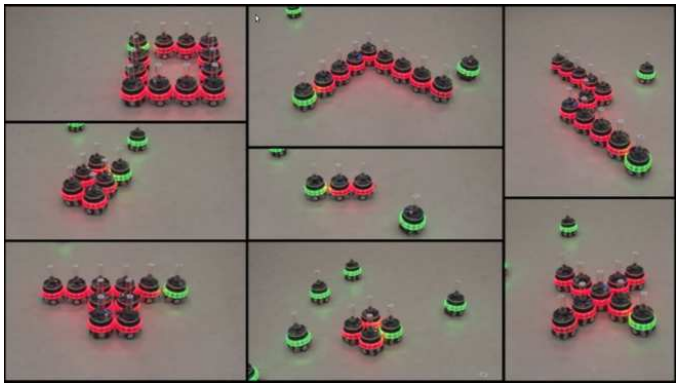


Figure 1.13. – Assemblage de robots homogènes [Mathews et al., 2012].

L'utilisation d'équipes homogènes montre des limites pour résoudre des problèmes complexes où des capacités de perception et d'action multiples sont nécessaires. Ainsi, depuis la fin des années 1990, les travaux se sont étendus aux systèmes hétérogènes, et notamment la coopération aéro-terrestre avec la preuve de concept de Alberto Elfes [Elfes et al., 1999] qui propose de piloter un véhicule terrestre grâce à des images aériennes. Plus récemment, des compétitions de robotique ont été organisées sur le thème de la coopération hétérogène. On peut citer le projet européen

Licence CC BY-NC-ND 3.0

ERL Emergency Robots, anciennement appelé *euRathlon,* qui a été inspiré par l'accident de Fukushima en 2011 (Figure 1.14). Le but est d'utiliser une équipe de robots aériens, terrestres et marins pour cartographier une zone et détecter des dangers critiques.

Figure 1.14. – Challenge européen de robotique hétérogène collaborative pour l'intervention en zone de danger.

La coopération aéro-terrestre peut prendre plusieurs formes. La vue aérienne du drone peut être utilisée pour guider le robot terrestre [Garzon Oviedo et al., 2012] [Guastella et al., 2017] ou aider à l'évitement d'obstacles [Fankhauser et al., 2016] [Li et al., 2016][Harik et al., 2015]. La vue d'ensemble du drone peut également être utilisée pour maintenir un déplacement en formation [Chaimowicz and Kumar, 2007] [Saska et al., 2014] [Guerin et al., 2015]. Dans le domaine de la cartographie, on peut trouver deux approches. La première approche vise à créer une cartographie pour assister le robot terrestre. Cela passe par une classification du terrain à partir de la vue aérienne [Sebastien Bosch, 2007] [J. Delmerico and D. Scaramuzza, 2017] ou par une détection des obstacles [Garzon et al., 2013]. La seconde approche vise à créer une cartographie commune à partir des deux points de vue aérien et terrestre [Mahendran et al., 2013] [Forster et al., 2013]. De manière générale, le point de vue aérien et le grand champ de vision en altitude sont les points forts du drone et font de lui un agent idéal pour la prise de décision. Cependant, l'autonomie énergétique des drones à voilure tournante est très souvent limitée. Une coopération constante limiterait la mission à une vingtaine de minutes avec des drones classiques. Pour pallier ce problème, le robot terrestre est parfois utilisé comme une station d'atterrissage mobile qui peut transporter et recharger le drone jusqu'à son prochain point d'exploration [Saska et al., 2012].

1.3.2. Architecture centralisée, décentralisée ou distribuée

La gestion de plusieurs agents nécessite une grande capacité de calcul pour gérer le comportement de chaque entité, leurs interactions, et leurs missions. Cette capacité de réflexion peut être répartie de plusieurs façons selon les besoins, le niveau

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

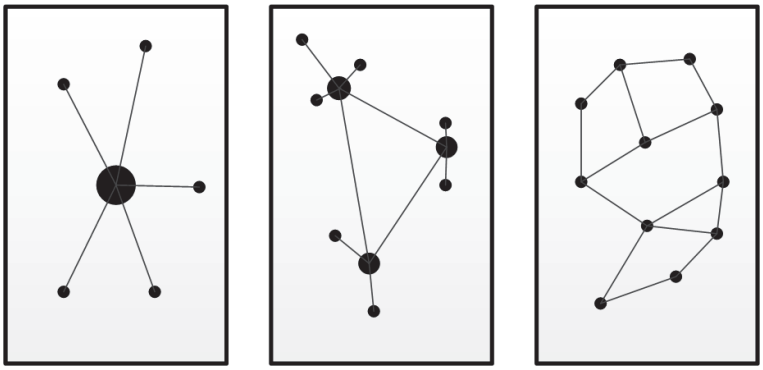


Figure 1.15. – Les trois types d'architectures : centralisée, décentralisée et distribuée.

d'autonomie de chaque élément et leur capacité de calcul embarquée. Ainsi, on peut distinguer trois types d'architectures : centralisée, décentralisée et distribuée.

— **Centralisée** : Un élément central, souvent une station fixe, est connecté à tous les éléments et réalise la plus grande partie de la planification. Toutes les communications passent par cette interface. C'est souvent le cas des systèmes dans lesquels un utilisateur intervient. Le nœud central, considéré comme le cœur du réseau, peut communiquer directement avec chaque élément, mais son rayon d'action est limité au rayon de communication. Cet élément central est un élément critique du système, car s'il disparait, plus rien ne fonctionne. Cette architecture est cependant très utilisée car elle est facile à mettre en place et permet de centraliser toutes les informations au même endroit.

— **Décentralisée** : Il n'y a pas d'élément central mais plusieurs groupes de systèmes qui communiquent entre eux. C'est le cas dans [Harik et al., 2015] et [L. Cantelli, 2013], où le drone est autonome. Il est asservi en position au-dessus du robot terrestre qui, lui, est piloté par une station à distance. Cette méthode permet, entre autre, un pilotage avec une vue à la troisième personne. C'est cette architecture qui est mise en place dans les systèmes de type leader/follower [Saska et al., 2014].

— **Distribuée** : Chacun des individus gère son comportement en fonction de son écosystème local. Le réseau peut être étendu et n'est pas limité par un unique élément. C'est le cas de la plupart des systèmes en essaim et des véhicules autonomes n'utilisant que des communications locales et des contraintes de visibilité pour maintenir une cohérence [James D, 2005]. Cette architecture est la plus robuste car, si l'un des éléments disparait, le système complet n'est que très peu affecté. Elle est également la plus complexe car, lorsqu'une information doit atteindre un élément distant, elle doit être propagée à travers les voisinages, ce qui induit des retards importants.

Licence CC BY-NC-ND 3.0

1.3.3. Planification externe ou autonome

Lorsqu'il s'agit de collaborations multirobots, il est nécessaire que les tâches de chacun des robots soient planifiées correctement et parfois re-planifiées en ligne. Cette étape est également appelée « allocation de tâches ». La planification d'une mission est une problématique complexe, qui demande des connaissances sur les robots utilisés comme : leur capacité de déplacement, les vitesses limites, et les moyens de perception et d'action. Cette planification peut être faite par l'Homme de manière centralisée, comme c'est le cas dans le projet Européen *TRADR*[2]. Ce projet a pour but de développer des techniques de collaboration multirobots dans le cadre de missions de sauvetage de longue durée en coordination avec les pompiers. Ce type d'architecture nécessite un haut niveau d'autonomie des robots pour qu'un seul utilisateur puisse piloter plusieurs robots. Les robots sont ainsi équipés d'algorithmes de SLAM et de navigation pour atteindre des positions définies par l'utilisateur expert [Dube et al., 2016].

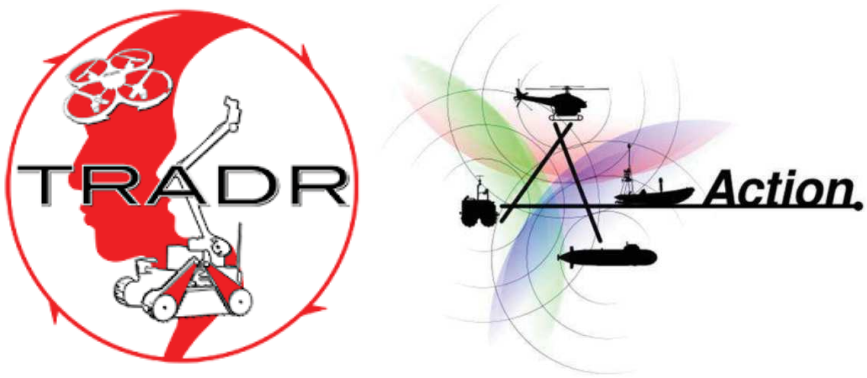


Figure 1.16. – Projets européens de robotique hétérogène.

Lorsque la mission doit être complètement autonome, la planification et la re-planification en ligne sont faites par des algorithmes d'allocation. De manière générale, l'objectif est de formuler une fonction qui projette chaque robot dans l'espace des tâches de la mission. Cette fonction doit prendre en compte le but de la mission, les tâches disponibles et les capacités de chaque entité [Riccio et al., 2015]. Cette formulation dépend entre autre de l'architecture de communication choisie, du nombre de robots et du scénario.

Le Plan d'Étude Amont (PEA) Action, un programme de recherche impliquant la DGA, l'ONERA et le LAAS, a permis le développement et l'implémentation de solutions pour la planification d'une mission avec des véhicules hétérogènes autonomes. Une architecture centralisée fournie à l'algorithme l'information de l'état de tous les

2. Long-Term Human-Robot Teaming for Disaster Response.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

robots. Cela permet d'avoir une auto-organisation de la flotte de robots où les robots échangent des données pour répartir les tâches individuelles. Parmi ces approches, on trouve des solutions à base d'enchères [Zlot and Stentz, 2006] ou d'allocations sous contraintes [Bechon, 2016].

1.4. Les défis à relever

Le SLAM est étudié depuis longtemps et des solutions robustes et matures ont été proposées. La question « Le SLAM est-il un problème résolu ? » peut donc être posée. Dans l'état de l'art [Cadena et al., 2016], Les auteurs expliquent que le SLAM est un problème dépendant du contexte et en particulier du triplé : robot utilisé (capacité de déplacement), capteurs utilisés (unique ou multiple, niveau de bruit) et environnement (petite ou grande échelle, intérieur ou extérieur, statique ou dynamique). Une fois ces trois paramètres définis, on peut chercher à répondre à la question. Dans le cas d'un robot au sol, en intérieur, avec un lidar, le problème peut être considéré comme résolu et des solutions clé en main existent. Le site *OpenSlam*[3] propose une liste d'algorithmes prêts à être utilisés tels que le GMapping pour un LIDAR ou l'ORB-SLAM pour une caméra. Néanmoins, il reste encore beaucoup de situations où les approches actuelles ne suffisent pas.

Les environnements changeants ou dynamiques

Les recherches sur le SLAM ont jusqu'à récemment fait l'hypothèse d'un monde statique. De petites variations pouvaient entrainer des erreurs importantes de cartographie ou de localisation. Ce problème peut être ressenti sur plusieurs échelles de temps. À court terme, lorsque l'environnement comprend des éléments dynamiques tels que des piétons ou des véhicules. Cela peut perturber l'estimation de la position de l'objet (mirage perceptuel) et il est important de pouvoir détecter ces éléments et ne pas les inclure à la carte de manière statique. Pour parvenir à résoudre ce problème, des approches ont été proposées comme, par exemple, l'utilisation d'une représentation adaptée qui peut être modifiée très rapidement [Lau et al., 2013]. Ce phénomène existe aussi à moyen terme, lorsque des éléments mobiles sont momentanément statiques, et risquent de ne pas être présents lors d'un autre passage. Dans ce cas, des approches sémantiques ont été proposées pour permettre de retirer des éléments possiblement mobiles [Drouilly et al., 2015]. Cela rend plus sûres les fermetures de boucles. Finalement, à très long terme, il faut pouvoir gérer des changements tels que la construction d'un nouveau bâtiment ou le changement d'apparence avec les saisons. Des études proposent des cartes acceptant des modifications ou des cartes enrichies de nouveaux éléments à chaque passage [Churchill and Newman, 2013]. Le SLAM dans un environnement dynamique reste un problème ouvert.

3. https ://openslam-org.github.io/

Licence CC BY-NC-ND 3.0

Une cartographie distribuée robuste

Lorsque plusieurs robots sont utilisés pour cartographier une zone plus rapidement ou avec différents points de vue, la fusion des cartes peut être un problème difficile à résoudre. En particulier lorsque les robots ont des capteurs différents et ne sont pas en communication permanente. Il faut arriver à garder une cohérence globale et à résoudre les problèmes de représentations multiples [Roldan et al., 2016]. Les référentiels et les échelles d'oservations peuvent varier d'un robot à l'autre et la collaboration devient très vite compliquée. Il faut pouvoir localiser les cartes et les robots les uns par rapport aux autres. Cela permet de déterminer les transformations relatives entre les cartes [Ferrao et al., 2017]. Des travaux récents utilisent des méthodes provenant du traitement d'images (descripteurs et transformations rigides) pour les appliquer à des grilles d'occupation. L'utilisation de grilles d'occupation permet une fusion immédiate des cartes. Cependant, cette méthode est difficilement applicable avec des représentations continues ou topologiques.

Les espaces à grandes échelles

Les cartes métriques denses portent beaucoup d'informations, peut être trop pour la cartographie d'environnements très larges. La taille mémoire est souvent très importante. Les représentations métriques éparses sont plus légères en mémoire mais sont souvent difficiles à interpréter par l'Homme et sont difficilement utilisables pour la planification de trajectoires. Des heuristiques d'implémentation comme la segmentation du stockage des données permettent d'utiliser des données dont le volume dépasse la mémoire vive de l'ordinateur, mais cette solution n'est pas fiable à long terme. Les représentations hybrides peuvent résoudre ce problème en utilisant un graphe ou chaque nœud représente une carte métrique locale [Schmuck et al., 2016]. Cette approche a plusieurs avantages. Premièrement, lors de fermetures de boucles, l'optimisation n'est appliquée que sur les cartes locales concernées, réduisant grandement le temps de calcul. Deuxièmement, l'utilisation d'un ensemble de cartes locales permet de répartir la carte globale sur plusieurs représentations de taille mémoire convenable. La carte globale peut ainsi être faite en temps réel à grande échelle toute en gardant un bon niveau de détails. Cependant, la taille mémoire globale de la carte est supérieure à une carte non partitionnée car les cartes locales se superposent à certains endroits et créent une redondance des informations. Il faut également stocker les connexions entre les cartes locales.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Conclusion

La coopération multirobots pour la création de cartes navigables est un problème complexe qui entrecroise les différents domaines que sont le SLAM, la représentation de l'environnement et les systèmes multirobots. Chacun de ces domaines possède ses propres problématiques et ses défis à relever.

Le problème de la localisation et de la cartographie simultanées est une brique essentielle à l'autonomie d'un robot. C'est un problème d'optimisation de la pose du robot et de la position des amers obtenus grâce à un traitement des données provenant des capteurs. La carte créée est une représentation de l'environnement perçue par des capteurs embarqués. Cette représentation peut être métrique, topologique ou hybride. Le choix du type de modèle utilisé est fait en fonction de l'environnement lui-même, des capacités de perception embarquées et de l'utilisation de la carte. Lorsque cette cartographie est faite par plusieurs robots, il faut prendre en compte la différence des points de vue et l'architecture du réseau de communication.

Dans ce manuscrit, nous nous plaçons dans le cadre de la coopération aéro-terrestre distribuée pour la création de cartes hybrides pour la navigation du robot terrestre. Les communications sont supposées parfaites. Nous proposerons une nouvelle approche qui tente de répondre aux défis présentés précédemment. Le système hétérogène aura pour but de créer une carte robuste et adaptée aux espaces dynamiques à grande échelle en partageant leurs connaissances provenant de points de vue différents.

Licence CC BY-NC-ND 3.0

2. Cadre de l'étude et représentation hétérogène

Sommaire

Introduction **40**
2.1. Travaux préliminaires **40**
2.1.1. Détection de l'espace navigable 41
2.1.2. Squelettisation de l'espace navigable 42
2.1.3. Création d'un graphe à partir d'un squelette 46
2.2. Cadre de l'étude et objectifs **47**
2.2.1. Limites de l'exploration terrestre 47
2.2.2. Objectifs d'une cartographie multirobots 48
2.2.3. Représentation topologique modulaire et évolutive . . . 48
2.3. Définition de la représentation hybride **49**
2.3.1. Structure de graphe hybride 49
2.3.2. Architecture logicielle 52
2.4. Carte dépendante de la source **54**
2.4.1. Formalisation de l'espace navigable 54
2.4.2. Carte aérienne 55
2.4.3. Carte terrestre 56
2.4.4. Carte satellite 58
Conclusion **59**

Introduction

Cette étude vise à définir un modèle de représentation commun aéro-terrestre. Cette représentation s'appuie sur des perceptions qui permettent d'acquérir des informations de sources hétérogènes pour le système coopératif. Elle s'appuie sur des travaux développés dans les domaines de la perception omnidirectionnelle et de la représentation topologique. Ces travaux ont donné naissance à des outils pour la navigation et la cartographie autonome pour un robot terrestre utilisant une caméra catadioptrique. Ces travaux préliminaires ont également permis de révéler les limites de l'exploration terrestre et l'intérêt d'une coopération aéro-terrestre. L'utilisation de plusieurs robots pour la cartographie apporte de nouvelles contraintes d'autant plus que les points de vue sont très différents. Les points de repères perçus par le robot terrestre tels que les portes et les panneaux ne sont pas visibles par le drone à cause de la perspective très prononcée et réciproquement. Nous avons défini une représentation modulaire et évolutive pour permettre la création d'un environnement commun pour la résolution de ce problème. Par la suite, nous présentons la stratégie choisie et son implémentation. Les avantages et inconvénients de cette représentation sont discutés puis les méthodes de création des cartes aériennes et terrestres sont détaillées.

2.1. Travaux préliminaires

Ces travaux se placent dans le cadre d'un robot mobile de type char équipé d'une caméra catadioptrique. Ce système permet une vision à 360° autour du robot en une seule image mais au prix de fortes déformations. L'image figure 2.1 montre le robot utilisé avec le capteur en haut du mat et un exemple d'image omnidirectionnelle acquise par ce capteur. Les travaux qui ont inspiré notre étude abordent deux sujets particuliers. Tout d'abord, la détection de l'espace navigable terrestre [Merveilleux et al., 2011], ensuite, le calcul de cartes de distances et la squelettisation. Ce dernier aspect permet la création d'une carte topologique [Marie et al., 2013].

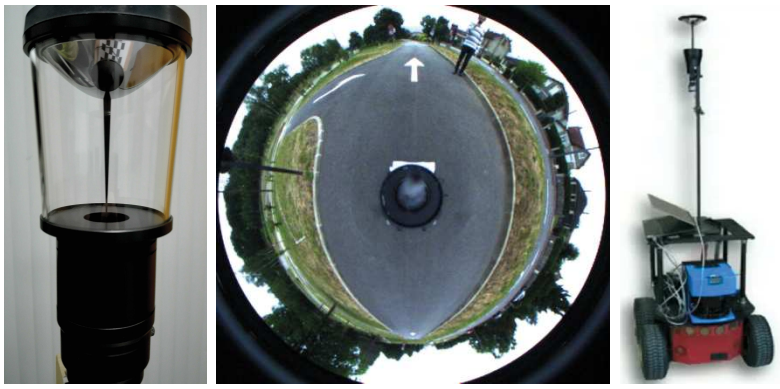


Figure 2.1. – Capteur catadioptrique et image omnidirectionnelle, et robot Pioneer équipé de son capteur.

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

2.1.1. Détection de l'espace navigable

Définition de l'espace navigable terrestre

L'espace navigable terrestre est la partie de l'environnement dans laquelle le robot terrestre peut se déplacer. Cette définition simple dépend du robot terrestre utilisé et de la mission. En effet, un robot « tank » avec des chaines peut se déplacer sur des terrains infranchissables (sable, cailloux, ...) par des robots équipés de roues. De même, la taille du robot peut transformer un petit trottoir en véritable mur.

Dans le cadre de ces travaux, les environnements considérés sont majoritairement des environnements structurés ou urbains. Le robot utilisé est capable de se déplacer sur la route. On ne prend pas en compte le trafic ou le sens de circulation.

Détection de l'espace navigable

Les travaux de thèse de Pauline Merveilleux [Merveilleux et al., 2011] ont permis de développer un outil de détection de l'espace navigable dans une image omnidirectionnelle. Il utilise la propagation d'un contour actif de type « *snake* » initialisé au centre de l'image catadioptrique. Le contour est discretisé en éléments qui se propagent pour minimiser une fonction énergie du contour. Cette fonction dépend de facteurs intrinsèques à la géométrie du contour et de paramètres liés au gradient local dans l'image. Cet algorithme permet d'avoir une extraction robuste de l'espace navigable, qui résiste aux obstacles mal classifiés. Ce contour actif est illustré dans la Figure 2.2.

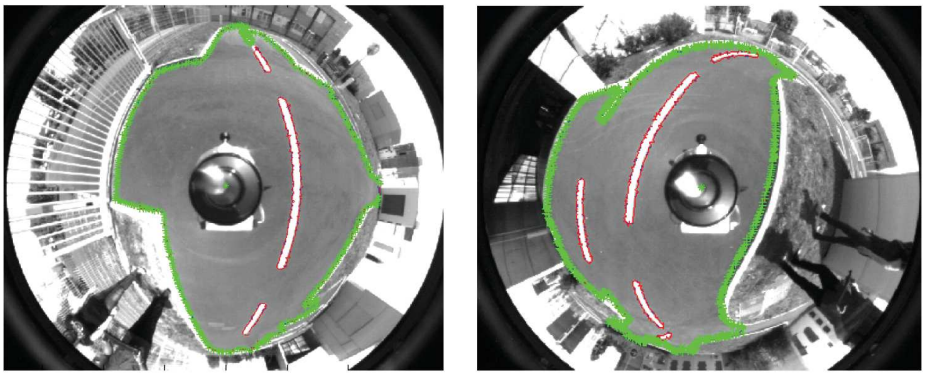


Figure 2.2. – Images omnidirectionnelles avec le contour actif propagé aux limites de l'espace navigable.

Le résultat de la détection de l'espace navigable est une image binaire, où le blanc correspond à l'espace navigable et le noir à l'espace non accessible (Figure 2.3).

Licence CC BY-NC-ND 3.0

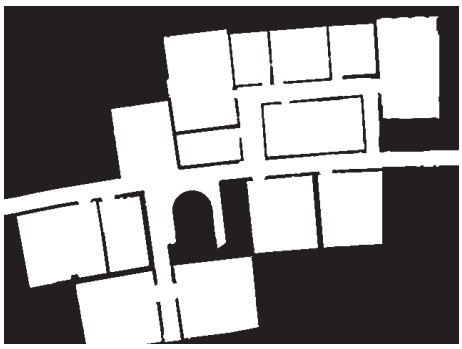


Figure 2.3. – Image binaire de l'espace navigable calculé à partir d'un plan d'un bâtiment. Le blanc correspond à l'espace navigable.

À la suite de ces travaux, l'équipe de recherche a travaillé sur la création de cartes topologiques de cet espace navigable terrestre, et notamment en utilisant un procédé de squelettisation.

2.1.2. Squelettisation de l'espace navigable

Dans cette section, nous allons aborder la définition du squelette et les techniques de squelettisation à l'origine de notre représentation de l'environnement.

Squelette et axe médian

L'axe médian $AM(\mathcal{X})$ a été introduit par Harry Blum en 1967 [Blum, 1967] comme une représentation compacte (descripteur) d'une forme $\mathcal{X}$. Il est défini comme l'ensemble des points à égale distance des frontières. Cette définition s'applique dans le cas d'un espace continu ou discret. Dans notre étude, nous calculons l'axe médian sur des images binaires, un exemple est présenté dans l'image ci-dessous (Figure 2.4).

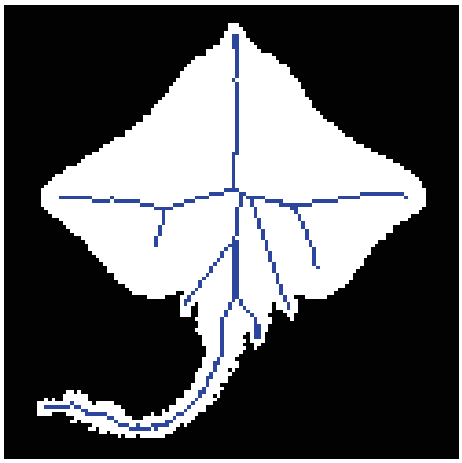


Figure 2.4. – Exemple d'axe médian pour une image de raie.

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

Le terme de « squelette » est apporté par Pfaltz en 1967 [Pfaltz and Rosenfeld, 1967]. Les deux termes (squelette et axe médian) désignent le même objet mathématique et sont souvent employés indifféremment, même s'il peut exister de légères variations selon les auteurs. Le terme de squelette sera davantage employé pour des objets physiques alors que l'axe médian sera plus utilisé pour des environnements. Par la suite, nous utiliserons principalement le terme d'axe médian. Ce dernier est en effet un bon descripteur de formes et est utilisé pour leur classification [James D, 2005] dans de nombreuses applications, à l'exemple de la reconnaissance de chiffres manuscrits [Kegl and Krzyzak, 2002].

L'axe médian possède un certain nombre de propriétés fondamentales. Les plus importantes sont les suivantes :

— **Finesse :** Dans le cas continu, le squelette est fin, son épaisseur est nulle. Dans le cas discret, cette propriété n'est pas toujours valide à cause des approximations et de la discrétisation. En deux dimensions, l'axe médian a donc une structure de graphe. Comme on peut le voir sur l'image (Figure 2.4).

— **Invariance en transformation affine :** Seules les modifications de forme impactent la forme du squelette. L'axe médian d'une forme est invariant par translation ou rotation aux erreurs de discrétisation près.

— **Homotopie :** L'axe médian est homotopique à la forme. On peut passer de l'un à l'autre de manière continue, ce qui a pour effet de garantir la connexité d'un graphe pour une forme connexe.

— **Idempotence :** L'axe médian d'un axe médian est lui-même. $AM(AM(\mathcal{X})) = AM(\mathcal{X})$.

— **Réversibilité :** On ne peut pas reconstruire une forme uniquement à partir de son axe médian. Cependant, en connaissant le rayon du disque inscrit maximal $r(x)$, on peut définir la transformée en axe médian : $TAM(\mathcal{X}) = \{(x, r(x)),\ x \in AM(\mathcal{X})\}$. Cet ensemble, points plus rayons, est quant à lui inversible.

— **Semi-continuité :** La transformation en axe médian n'est que semi-continue. En effet, une faible modification de la forme peut entrainer de grands changements dans l'axe médian.

— **Positionnement :** L'axe médian se trouve sur les axes de symétrie de la forme mais également le plus loin des frontières. Ce qui en fait un outil pertinent en robotique pour la génération de trajectoires sûres [Marie et al., 2018].

Le squelette d'une forme dans une image peut être construit de plusieurs façons. Blum [James D, 2005] utilise une approche par front d'onde aussi appelée « feu de prairie », qui consiste à créer une onde partant des frontières. Les zones d'autocollision de l'onde sont les éléments de l'axe médian. Une autre approche procède par amincissement de forme où on utilise des transformations morphologiques pour

Licence CC BY-NC-ND 3.0

réduire progressivement une forme à une épaisseur nulle. Dans notre cas, nous utilisons une approche fondée sur la carte des distances, où l'axe médian correspond aux crêtes de la transformée en distance.

Carte des distances et diagramme de Voronoï

La carte des distances (*Euclidian Distance Transform*) d'une forme est calculée à partir d'une image binaire. La carte des distance de la figure 2.3 est représentée ci-dessous (Figure 2.5). L'axe médian peut être défini en géométrie comme l'ensemble des points les plus éloignés des frontières. Dans le domaine du traitement d'image, cela revient à calculer l'ensemble des pixels situés sur les « crêtes » de la carte des distances.

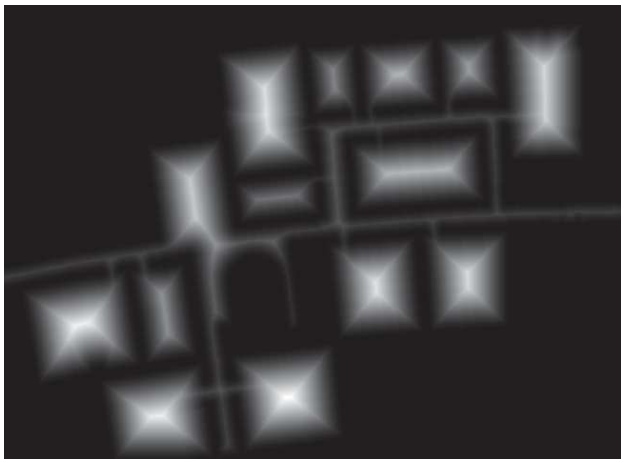


Figure 2.5. – Carte des distances d'une image binaire. La nuance de gris montre la distance aux frontières.

Dans cette image, le dégradé de gris représente la distance aux bords. On peut ainsi voir apparaitre le squelette des formes en suivant les parties les plus blanches (crêtes). Pour un pixel x, on définit ses voisins $\mathcal{V}_x$ comme l'ensemble des pixels à une distance de 1 ($\mathcal{V}_x = \{y \,|\, d(x,y) = 1\}$). On définit également sa distance au bord $dt(x)$ comme la distance entre x et le point de l'espace non franchissable le plus proche $ft(x)$. On appelle $ft(x)$ la projection de x. Ce pixel x appartient à l'axe médian si un de ses voisins a une distance au bord inférieure à la sienne et que leur projection sont distinctes. On a alors :

$$x \in MA \iff \exists y \in \mathcal{V}_x \mid \begin{cases} ft(x) \neq ft(y) \\ \& \\ dt(y) \leqslant dt(x) \end{cases}$$

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

l'axe médian a fait l'objet de nombreuses recherches, notamment pour ses propriétés de compression [Farazmanesh and Tavakoli, 2018] et son utilisation pour la classification des images [Sebastian and Kimia, 2001]. Cependant, cette définition classique ne permet pas d'avoir un squelette lisse pour la plupart des environnements. De nombreuses branches parasites peuvent en effet apparaitre du fait de perturbations des bords, même faibles. Il faut donc utiliser un critère de filtrage pour enlever ces branches parasites (Figure 2.7).

La construction de l'axe médian peut se faire en utilisant des diagrammes de Voronoï (VD) qui serviront de bases à l'extraction de graphes. Ce diagramme de Voronoï est un pavage de l'espace où les cellules représentent les zones les plus proches de points appelés « germes ». Ainsi, beaucoup d'algorithmes de squelettisation portent le nom de graphe de Voronoï (VG). On peut notamment citer l'*Extended VG* [Beeson et al., 2005] qui garde un squelette proche des frontières lorsque les espaces sont trop grands pour les perceptions locales du robot. Le *Generalized VG* [Choset and Nagatani, 2001] montre les premières utilisations des cartes topologiques dans un algorithme de SLAM. Le *Saturated-GVG* [Tao et al., 2011] ajoute au GVG un suivi de murs permettant une construction robuste de la carte gérant les conflits de représentations grâce à des hypothèses multiples. Dans nos travaux, nous avons utilisé le *Delta Medial Axis* (DMA) proposé par R. Marie [Marie et al., 2013] qui permet d'obtenir un axe médian filtré grâce à un unique paramètre δ qui correspond à la largeur acceptable des passages pour un robot.

Delta Medial Axis

Cet algorithme utilise la carte des distances pour extraire le squelette. Le critère de sélection d'un point du squelette est modifié pour faire intervenir le paramètre de taille limite δ. La condition d'appartenance au squelette est reformulée ainsi :

$$x \in DMA \iff \exists y \in \mathcal{V}_x \mid \begin{cases} & \exists z \in [ft(x), ft(y)] \mid dt(z) > \delta \\ \& & \\ & dt(y) \leqslant dt(x) \end{cases}$$

Étant donné deux points voisins x et y, la première condition exprime que x appartient au squelette s'il existe un point arbitraire z sur le segment $[ft(x), ft(y)]$ où un disque de rayon δ est complètement contenu dans la forme. Cette condition évite d'inclure des points liés à des approximations de formes discrètes ou à des données bruitées. Quatre situations où l'on peut voir comment est calculé le paramètre δ sont présentées dans l'illustration suivante (Figure 2.6). La seconde condition assure qu'un seul point est affecté au squelette. Si x et y sont à égale distance du bord, les deux sont inclus en tant que points du squelette.

Licence CC BY-NC-ND 3.0

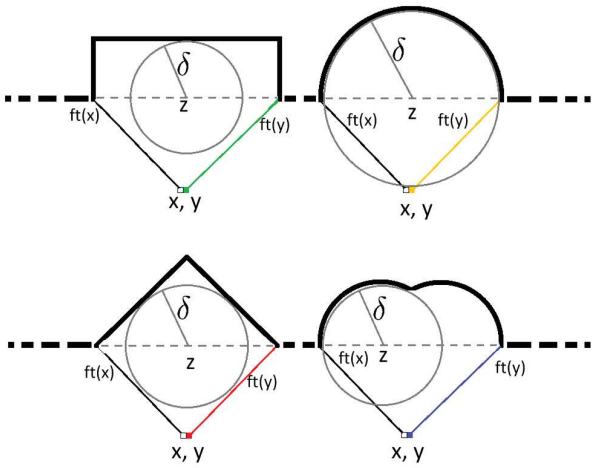


Figure 2.6. – Illustration de la condition du DMA. Le point x n'appartient à l'axe médian que si δ est supérieur au paramètre δ limite.

Cet algorithme a été appliqué pour la squelettisation de l'espace navigable, il peut ainsi servir de support aux chemins optimaux en terme d'éloignement aux obstacles [Marie et al., 2018]. Le résultat de cet algorithme, pour une image représentant le plan d'un bâtiment, est présenté dans la figure 2.6. À gauche, on peut voir l'axe médian brut et à droite, la version filtrée obtenue en utilisant le DMA. On peut ainsi clairement voir la topologie de l'espace. Les points du squelette ainsi créés portent la distance aux obstacles donnée par la carte des distances.

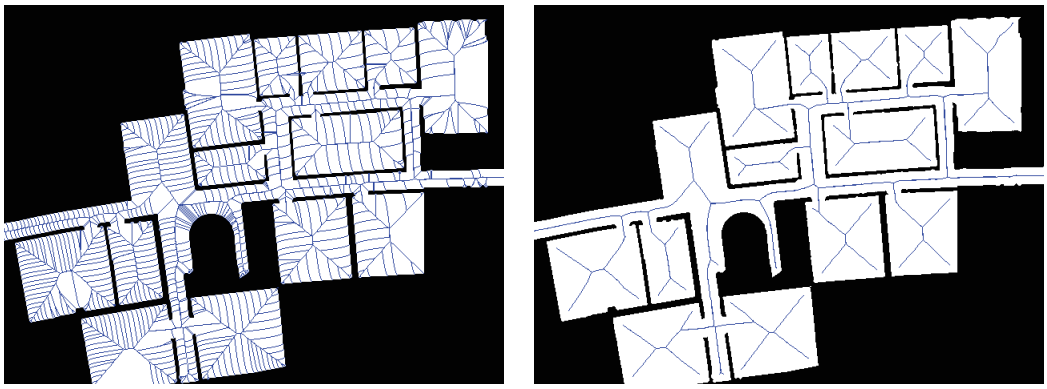


Figure 2.7. – Illustration de l'axe médian pour une forme. À gauche, axe médian sans filtrage et à droite, le *delta medial axis*.

2.1.3. Création d'un graphe à partir d'un squelette

Un algorithme a été développé pour pouvoir créer une carte topologique à partir du squelette calculé dans une image binaire de l'espace navigable. Pour extraire le

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

graphe à partir du squelette, l'algorithme procède en deux étapes. Premièrement, les intersections et les impasses sont détectées pour définir les nœuds du graphe. La position du nœud est la position du point dans l'image. Ensuite, les branches du squelette sont parcourues pour créer les arêtes. Cette carte topologique est pertinente pour le calcul de chemin.

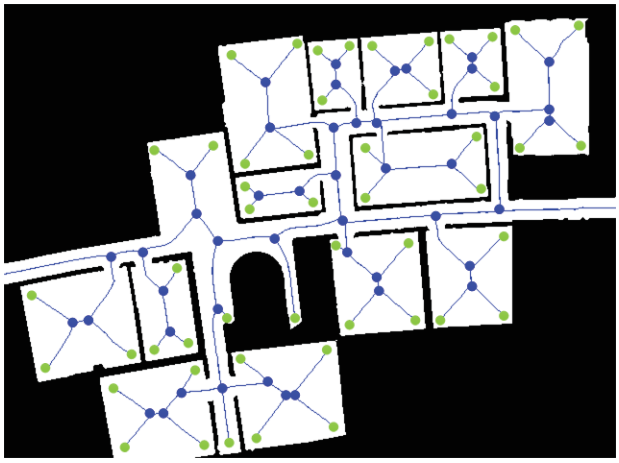


Figure 2.8. – Graphe créé à partir de l'axe médian. Les nœuds en vert sont les impasses et les nœuds bleu sont les intersections.

La figure 2.8 montre la carte topologique construite à partir de l'axe médian précédant. Les bords de l'image ne sont pas considérés comme des obstacles ainsi, des chemins peuvent se prolonger en dehors de l'image sans créer de nœuds impasse ni d'arête (à gauche et à droite dans la Figure 2.7).

2.2. Cadre de l'étude et objectifs

Ces travaux ont permis de développer des outils pour la navigation et la cartographie pour un robot terrestre, mais ils ont également permis d'en percevoir des limites.

2.2.1. Limites de l'exploration terrestre

Les limites de l'exploration uniquement terrestre mises en avant par ces études sont doubles. Premièrement, le champ de vision du robot est limité. En effet, la caméra du robot a un champ de vision utilisable assez restreint, malgré l'utilisation d'un objectif catadioptrique omnidirectionnel. Il est également limité par les obstacles qui l'entourent. Deuxièmement, dans le cadre des études précédentes, le robot utilisait son odométrie pour la création des cartes en environnement intérieur. Ainsi, les cartes créées souffrent de déformations dues à la dérive de l'odométrie.

Licence CC BY-NC-ND 3.0

Ces limites s'ajoutent aux limitations physiques du robot utilisé : une vitesse faible et une mobilité restreinte qui rendent le processus d'exploration très long.

Les travaux de R. Marie [Marie, 2014] ont fait ressortir la similitude entre les cartes topologiques créées par le robot terrestre et celles obtenues par imagerie aérienne. Ces deux observations ont donné naissance à l'étude proposée ici.

2.2.2. Objectifs d'une cartographie multirobots

La définition d'une cartographie commune pour un système hétérogène est une problématique complexe. Nous souhaitons utiliser un drone pour obtenir une carte navigable pour le robot terrestre plus rapidement et avec moins de distorsions qu'en utilisant un robot terrestre seul. Cependant, plusieurs facteurs rendent le procédé compliqué. Les points de vue et les capteurs embarqués sur les robots ne sont pas les mêmes. Le robot terrestre dispose d'une caméra omnidirectionnelle et d'un lidar alors que le drone dispose d'une caméra perspective. Les points de vue et la taille des champs de vision sont également très différents. Les référentiels des cartes construites ainsi que leur échelle ne sont pas forcément les mêmes non plus.

Pour pouvoir comparer et fusionner les informations provenant de sources hétérogènes, nous souhaitons utiliser un type de représentation unifié pour les cartes créées par ces deux systèmes.

L'un des objectifs est donc de définir un modèle de l'environnement utilisable par un robot pour la planification de chemin, la navigation locale et la localisation. Ce modèle doit pouvoir supporter des informations de plusieurs types et de plusieurs sources. Il doit, bien sûr, être adapté aux capteurs du drone et à ceux du robot terrestre.

L'autre objectif est de développer un algorithme permettant la mise en correspondance et la fusion des cartes utilisant cette représentation.

2.2.3. Représentation topologique modulaire et évolutive

Les travaux précédents ont ouvert la voie vers une cartographie se concentrant sur la forme de l'espace navigable. Cela permet d'avoir des cartes avec une taille mémoire réduite, utilisables sur de grands environnements. Cet avantage limite cependant la représentation de l'environnement, puisque ce qui ne fait pas partie de l'espace navigable n'est pas représenté. La carte pourrait être enrichie d'informations métriques ou sémantiques. Dans [Marie, 2014], cette représentation a été étendue pour intégrer des signatures visuelles servant notamment à la localisation et à la détection de fermetures de boucle.

Comme expliqué dans le chapitre précédent, les représentations purement métriques ou topologiques sont rarement suffisantes pour offrir au robot une forte autonomie

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

décisionnelle. Une utilisation conjointe de ces représentations permet de répondre à ce problème. La structure topologique autorise la planification de trajectoires et les informations métriques permettent une identification des lieux pour la localisation. Ce choix est également motivé par les capacités de hiérarchisation, de compacité et de modularité de cette représentation.

Pour que ce modèle soit adaptable sur différents types de robots, nous avons utilisé une architecture logicielle modulaire. Ceci nous permet l'ajout de données différentes et une modification rapide et simple de la méthode de fusion. En effet, les nœuds et les arêtes de la carte hybride peuvent être enrichis de nombreuses informations utiles. Ces informations peuvent alors être utilisées dans le processus de fusion des cartes.

2.3. Définition de la représentation hybride

2.3.1. Structure de graphe hybride

Nous avons fait le choix de représenter l'espace navigable en utilisant un modèle hybride topologique, métrique et sémantique.

Le modèle choisi est donc un graphe $G = \{N, E\}$ constitué d'une liste de nœuds N et d'arêtes E. Les nœuds sont les lieux distinctifs de l'environnement définis par les intersections et les impasses de l'axe médian. Les arêtes sont les connexions entre eux, comme proposé par Kuipers [Kuipers, 1978]. Ces deux éléments sont enrichis de plusieurs types d'informations appelées "Données".

Nœud
- *Source*
- *Position*
- *Rayon*
- Liste des arêtes connectées :
  - *Index*
  - *Angle d'incidence*
  - *Longueur*
  - *Largeur*
  - *Courbe*
- ...

Arête
- *Source*
- *Longueur*
- *Largeur*
- *Courbe*
- ...

Le schéma (Figure 2.9) présente une illustration de ce graphe hybride. On peut y voir six nœuds avec des rayons différents (en vert) et des angles d'incidence (en rouge). Il y a également sept arêtes représentées par des traits noirs, les courbes des chemins correspondants sont en pointillés bleus.

Cette représentation est construite en créant un graphe à partir du squelette de l'espace navigable. Les informations métriques sont extraites de la carte des distances

Licence CC BY-NC-ND 3.0

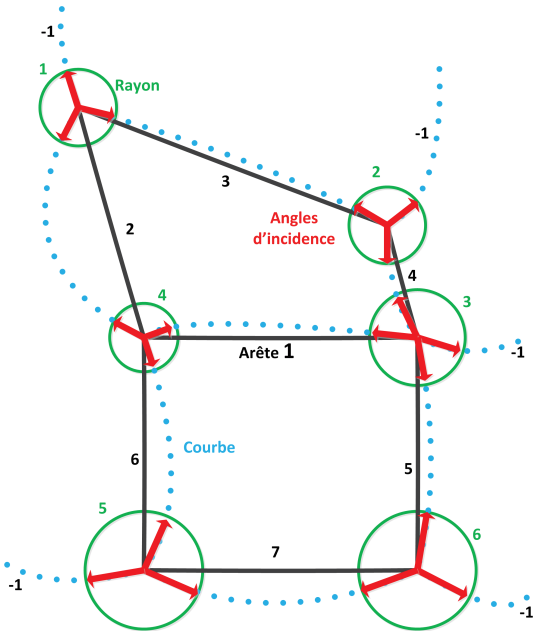


Figure 2.9. – Représentation hybride proposée.

lors de la création du graphe. Chaque nœud/arête possède un index permettant de l'identifier. Pour un nœud donné, sa position est décrite par la position du point dans l'image. Le rayon décrit la transformée en distance en ce point (et donc la distance à l'obstacle le plus proche). La donnée « source » d'un nœud/arête permet d'identifier le type de chaque élément (aérienne, terrestre, ...). Un nœud comprend une liste des arêtes incidentes avec l'angle d'incidence et les informations métriques des chemins correspondants. Un nœud peut être connecté à un chemin non exploré, il n'y alors pas d'arête (l'index vaut -1). Pour une arête donnée, la courbe est définie par une liste de points échantillonnés tout le long de la branche du squelette qu'elle décrit. L'information de largeur est la liste des transformées en distance des points de la courbe.

Dans la carte de l'environnement intérieur considéré précédemment (Figure 2.10), on peut voir l'échantillonnage des chemins grâce aux points bleus représentant leur courbe. Les cercles verts représentent les rayons des nœuds et leur index est indiqué en rouge. Selon les robots et les traitements utilisés, d'autres "données" peuvent être ajoutées aux nœuds et arêtes comme par exemple des informations sémantiques ou une position GPS. Nous présenterons des exemples concrets dans le chapitre 4 (expérimentations).

La carte (Figure 2.11) montre un exemple de données sémantiques pouvant être portées par le graphe. Les nœuds et les arêtes portant l'information « salle » sont en violet et les « couloirs » sont en jaunes. La manière de détecter et de classifier ce type d'information n'est pas l'objet principal de cette thèse. Dans le cadre de notre étude, nous avons utilisé une méthode fondée sur l'analyse du degré et du voisinage

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

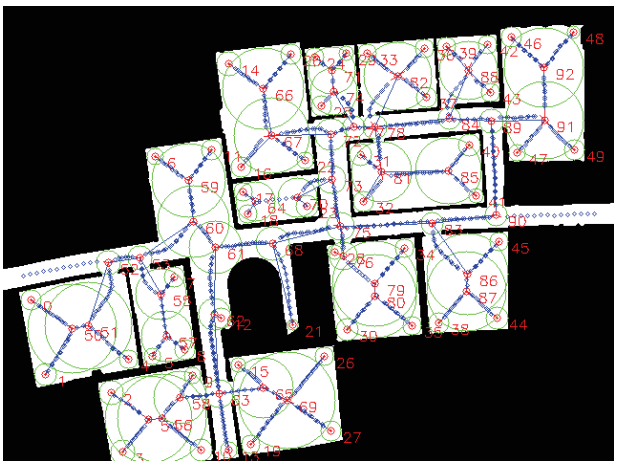


Figure 2.10. – Carte hybride topométrique d'un environnement intérieur.

des nœuds. Les nœuds présentant une seule arête (impasse) dont la largeur se réduit rapidement sont caractéristiques des « coins » des salles. Les coins des salles sont reliés à des nœuds « salle » et les « couloirs » correspondent aux arêtes dont la largeur varie peu le long du chemin. Ce raisonnement permet de déterminer le type de lieu auquel appartiennent les éléments du graphe.

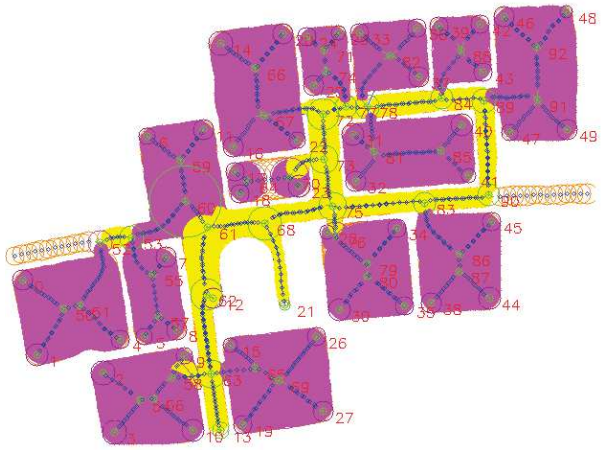


Figure 2.11. – Données sémantiques portées par la carte hybride. Les régions en violet correspondent aux salles, et celles en jaune aux couloirs.

Notons que d'autres méthodes sont envisageables, notamment les réseaux de neurones entrainés comme dans [J. Delmerico and D. Scaramuzza, 2017]. L'utilisation de ces informations n'a pas été plus approfondie dans le cadre de cette étude, mais un paragraphe lui sera dédiée dans les perspectives de ce manuscrit.

2.3.2. Architecture logicielle

La représentation définie dans cette étude correspond à la norme des représentations 2D rédigée par IEEE Robotics and Automation Society [IEEE, 2015]. Des spécifications d'implémentation permettent l'adaptabilité de cette carte.

Plusieurs types de "Données"

Afin de pouvoir faire varier le contenu des éléments du graphe sans avoir à redéfinir la structure, nous avons choisi d'utiliser une architecture modulaire. Les objets (*Classes*) du graphe, des nœuds et des arêtes sont les mêmes, quelle que soit la source d'information. Les nœuds et les arêtes contiennent une liste de "Données" sans particularité. Ces "Données" sont définies par une *classe mère virtuelle* qui ne contient qu'un seul élément : le type de donnée en format texte. Les *données filles* sont définies par héritage de cette classe mère et contiennent les informations qu'elles désignent. La position d'un nœud est ainsi définie dans une classe "Coordonnée" avec le type du même nom et les informations X et Y. La donnée "Sémantique" contient, entre autre, les informations sur la source. Cette architecture (Figure 2.12) permet de facilement créer une nouvelle donnée adaptée aux capteurs utilisés comme par exemple des données GPS, sans avoir à modifier le graphe et ses éléments.

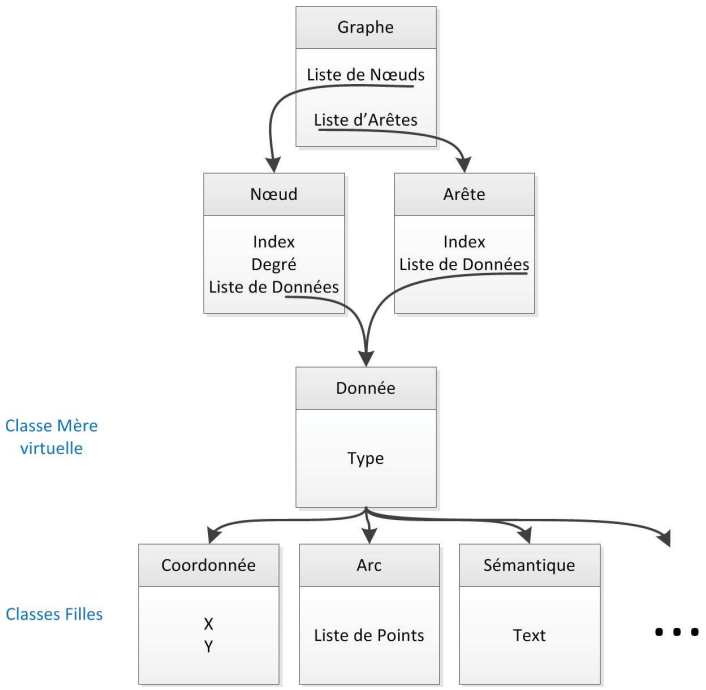


Figure 2.12. – Diagramme de l'architecture logicielle.

Des outils sont également définis pour générer automatiquement l'affichage et la sauvegarde de ces données.

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

Comparaison du volume mémoire

Cette représentation est adaptée à la modélisation des environnements structurés (réseau routier, intérieurs). L'approche adoptée a besoin de frontières pour calculer le squelette. Elle peut donc être limitée dans les environnements peu structurés ou complètement vides. Une autre méthode de construction du graphe serait alors nécessaire ;

La carte métrique de l'environnement peut être reconstruite en « gonflant » le squelette. La qualité et la précision de cette reconstruction dépendent de l'échantillonnage utilisé, de la précision de la squelettisation (les régions dont la distance au bord est inférieure à δ ne seront pas reconstruites) et de la technique de gonflement utilisée. Dans ce sens, cette représentation joue le rôle de compression de données. Dans l'illustration (Figure 2.10), l'image originale utilise 105ko alors que le graphe n'utilise que 51ko .

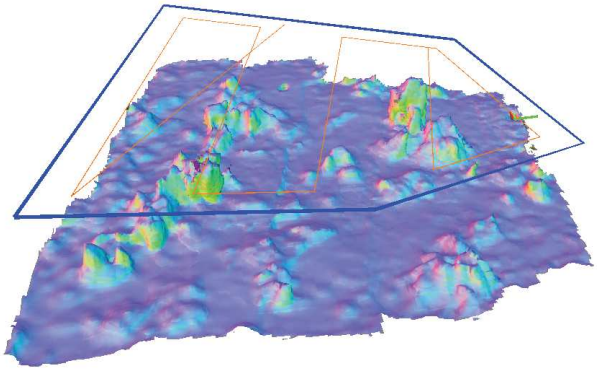


Figure 2.13. – Cartographie 3D de l'espace vert avec la trajectoire du drone.

Des tests de cartographie ont été effectués dans un espace vert ouvert. La zone fait 18m de large et 19m de long, elle a été survolé par un drone équipé d'une pair stéréo afin d'obtenir des images de profondeur (Figure 2.13). Le tableau suivant présente la taille mémoire des différents types de cartes.

Type de représentation (extension)	Taille du fichier
PolygonMesh (.ply)	10500 Ko
Nuage de Points (.pcd)	7700 Ko
GridMap traversabilité (.rosbag)	4000 Ko
GridMap (.rosbag)	610 Ko
Carte hybride (.txt)	30 Ko
Image de traversabilité (.png)	5 Ko

Table 2.1. – Volume mémoire en fonction du type de représentation.

La taille des fichiers est bien cohérente avec la quantité d'information contenue. Le PolygonMesh contient le nuage de points plus les surfaces. La GridMap est une version allégée du nuage de points puisqu'elle est une projection en 2D de l'altitude des points. La carte hybride ne contient que l'information utile pour le robot terrestre, c'est à dire la position des points, la largeur des voies navigables et d'autres informations relatives au graphe. Enfin, l'image de traversabilité utilise très peu de ressources car c'est une image binaire compressée. On note que le volume mémoire des fichiers augmente avec la taille des espaces à explorer.

2.4. Carte dépendante de la source

Cette représentation de l'environnement peut être construite par tout type de robot capable de détecter l'espace navigable. Néanmoins, le processus de création dépend des modalités de perception et de son champ de vision.

Comme le montre les images précédentes (Figure 2.3→Figure 2.10), ce traitement peut être appliqué aux plans de bâtiments pour construire la carte d'un environnement intérieur. Dans le cas de cette étude, les espaces navigables sont obtenus à partir d'images (terrestres ou aériennes) projetées dans le plan du sol (*orthoimage*).

2.4.1. Formalisation de l'espace navigable (cartes de traversabilité)

La segmentation et la classification d'images est une problématique complexe qui fait encore l'objet de recherches. Dans [J. Delmerico and D. Scaramuzza, 2017], la solution développée est d'utiliser l'imagerie 3D ainsi qu'un réseau de neurones entrainé pour classifier le type de sol. Dans le cadre de cette thèse, nous avons étudié plusieurs méthodes en fonction du contexte applicatif et du capteur visuel utilisé. Dans les environnements urbains où un contraste élevé est détectable entre les routes et le reste de l'environnement, nous avons utilisé une segmentation couleur et un filtrage morphologique adapté. Un exemple de résultat de segmentation est présenté dans l'illustration suivante (Figure 2.14). Cette méthode qui fonctionne bien sur des environnements simulés n'est cependant pas toujours adaptée au monde réel. Les ombres sont souvent détectées comme des routes. Des tests de cette méthode ont été réalisés en simulation et en environnement réel. Les résultats sont présentés dans le chapitre 4.

Une autre solution que nous avons exploitée dans le cadre de perceptions catadioptriques est l'extraction de l'espace navigable par propagation de contours actifs. Le principe consiste à « gonfler » une forme, à partir du centre de l'image (position du robot), pour trouver les limites de l'espace navigable [Merveilleux et al., 2011]. C'est la méthode utilisée pour la détection de l'espace navigable dans l'image omnidirectionnelle du robot terrestre et pour la création de la carte en utilisant le plan d'un

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

bâtiment. Ce procédé est aussi fondé sur le contraste entre la route et les obstacles, mais l'approche locale et le champ de vision limité du robot terrestre permet de réduire les risques d'erreurs.

Enfin, une approche utilisant la carte d'élévation obtenue par caméra stéréoscopique a également été utilisée. La carte de traversabilité est extraite à l'aide d'un seuillage sur l'élévation ainsi que l'inclinaison du sol. Cette approche est présentée en détail dans le chapitre 4.

2.4.2. Carte aérienne

Un drone équipé d'une caméra perspective pointée vers le bas peut percevoir une large zone de l'espace navigable terrestre. Au moyen du traitement d'image précédent (sous-section 2.4.1), on extrait la route pour définir la carte de traversabilité et créer la carte hybride. Les images suivantes (Figure 2.14) présentent une vue aérienne obtenue depuis le drone (à gauche) et la carte générée à partir de cette image (à droite). Lorsque le drone ne vole pas assez haut pour percevoir toute la zone d'intérêt, on peut utiliser un algorithme de mosaïque d'image ou une complétion incrémentale de la carte grâce à la méthode proposée dans [Renaudeau et al., 2018]. Cette méthode sera détaillée dans le chapitre 4.

Figure 2.14. – Segmentation et cartographie de l'image aérienne. (gauche) Image obtenue dans le simulateur 4Dvirutaliz sur le site Pavin. (droite) L'espace navigable extrait est en blanc et la carte hybride est superposée.

La figure 2.15 montre la carte hybride globale calculée pour l'environnement Pavin. Les cartes de cet environnement serviront d'exemples dans la suite de ce manuscrit. L'ensemble des expérimentations et des environnements utilisés seront présentés dans le chapitre 4.

Licence CC BY-NC-ND 3.0

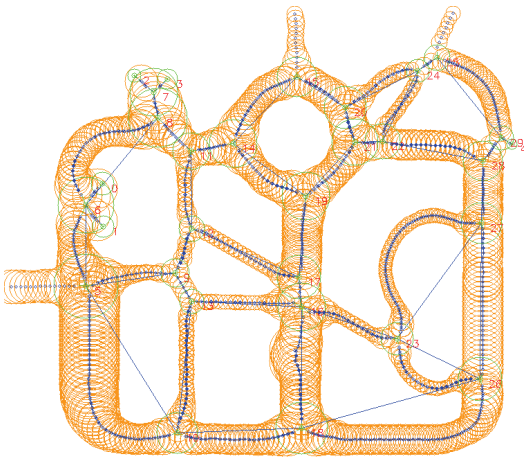
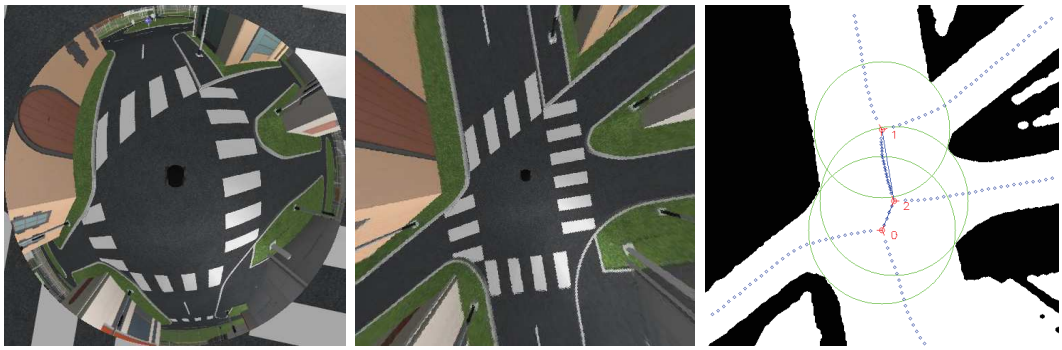


Figure 2.15. – Carte globale du site Pavin créée par le drone.

2.4.3. Carte terrestre

Le point de vue d'un robot terrestre est beaucoup plus proche du sol et le champ de vision plus réduit que celui du drone. Il ne permet pas d'avoir une vision globale de l'environnement. Le robot terrestre construit incrémentalement sa carte à partir de sa perception locale et des données odométriques. L'approche utilisée est un SLAM visuel monoculaire à partir d'une caméra omnidirectionnelle catadioptrique embarquée sur le robot. L'image obtenue est transformée en *bird-eye-view* par projection de l'image dans le plan du sol à l'aide des caractéristiques géométriques du miroir de la caméra. On limite cette vue à une certaine distance du robot car la qualité de la perception diminue avec cette distance. On peut ensuite calculer la carte hybride locale de cette orthoimage. Les images présentées ci-dessous (Figure 2.16) montrent successivement l'image omnidirectionnelle obtenue par le robot, la bird-eye view et la carte locale générée pour cet exemple.



1) Image omnidirectionnelle 2) Bird-eye view 3) Carte hybride locale

Figure 2.16. – Cartographie terrestre locale par vision catadioptrique.

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

La carte globale est construite de manière incrémentale lors de l'exploration du robot. Les nœuds ne sont ajoutés à la carte globale que lorsque le robot passe par les intersections. Les arêtes sont créées lorsque le robot passe d'un nœud à un autre. Le chemin d'une arête est créé en enregistrant le point du squelette le plus proche de la position du robot ainsi que sa distance au bord. Cela permet d'éviter les déformations liées aux erreurs de segmentation aux bords de la bird-eye view. En effet, la projection de l'image omnidirectionnelle en orthoimage fait que la qualité varie au sein de l'image. Le centre préserve mieux les distances que les bords déformés par la projection. Un des avantages de cette cartographie est qu'elle ne nécessite pas la navigation sur le squelette pour créer la carte hybride. Le résultat de la cartographie pour un chemin particulier dans l'environnement Pavin est présenté ci-après (Figure 2.17). On peut voir clairement que cette carte est déformée. Cela est dû aux erreurs d'odométrie qui font dériver le positionnement des nœuds au fur et à mesure du déplacement. De plus, le sol n'étant pas plat à certains endroits, cela induit des erreurs lors de la projection de l'orthoimage. Une méthode de fermeture de boucle est mise en place pour relocaliser le robot. Les perceptions locales sont comparaient aux nœuds de la cartes pour déterminer si le robot traverse une intersection déjà explorée. Pour mettre en avant la différence entre les cartographies aérienne et terrestre, aucun ajustement n'est calculé sur la carte.

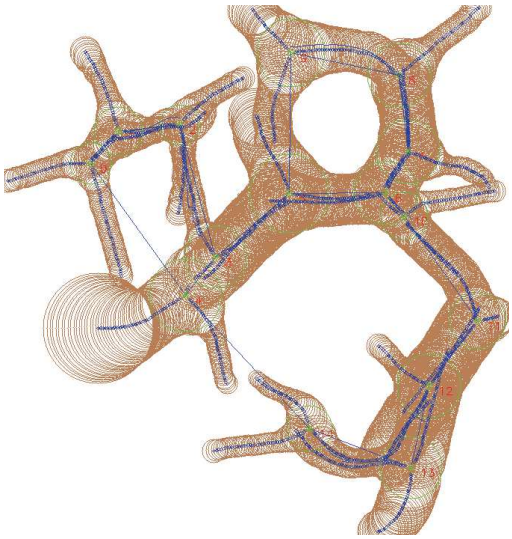


Figure 2.17. – Carte du site Pavin créée par le robot au sol. On peut notamment voir les déformations dues à l'odométrie sur certaines zones.

Cette cartographie terrestre est loin d'être parfaite mais l'intérêt d'une cartographie coopérative n'en est que plus visible. Le but de la cartographie aéro-terrestre est bien de corriger cette carte en exploitant celle construite par le drone.

Licence CC BY-NC-ND 3.0

2.4.4. Carte satellite

Dans le cas de l'utilisation du réseau routier comme espace navigable, une autre source d'information est accessible. Ce sont les bases de données en ligne. Les réseaux routiers, ainsi que d'autres informations potentiellement utiles, sont disponibles grâce à des outils tels que OpenStreetMap(OSM).

OpenStreetMap est un projet créé en 2004 par Steve Coast. Il a pour but de constituer une base de données géographiques libre du monde. Il est construit sur la collaboration de tous les utilisateurs volontaires (comme un wiki).

La représentation géographique contient plusieurs types d'informations (couches) :

- Les Points (arrêt de bus, arbres, boutiques, ...)
- Les Routes (routes, chemins, pistes cyclables, ...)
- Les Méta-routes (lignes de bus, métro, lignes électriques, ...)
- Les Zones (forêts, bâtiments, propriétés privées, ...)
- Les images aériennes (données raster)

Ces données sont libres d'utilisation. Les fichiers téléchargeables sont au format XML avec l'extension « .osm », et sont utilisables par la plupart des logiciels de Système d'Information Géographique (SIG). Ils contiennent le tracé des routes, la position de bâtiments et des informations sémantiques telles que le nom et le type de route. Grâce à une méthode simple, il est possible de créer la carte hybride à partir de ce fichier. Les détails sur les données OSM ainsi que le processus de création de la carte hybride sont disponibles en annexe B. L'image suivante (Figure 2.18) montre la carte de l'environnement Pavin obtenue grâce à cette base de données.

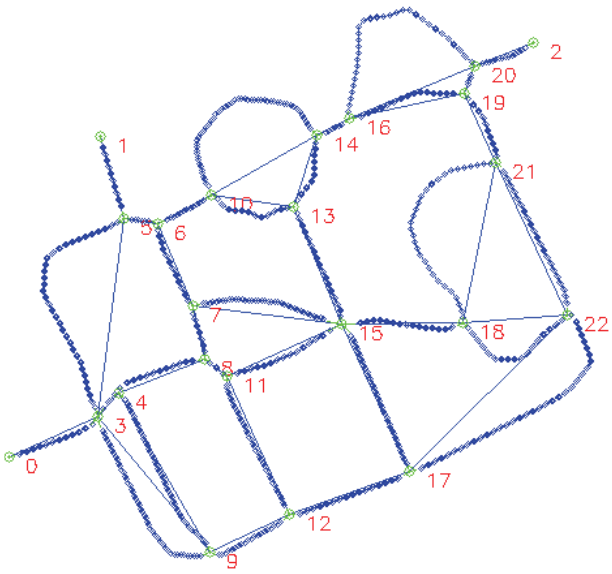


Figure 2.18. – Carte hybride construite à partir des données de OpenStreetMap.

Brice Renaudeau - Thèse de doctorat - Université de Limoges
Licence CC BY-NC-ND 3.0

Conclusion

Le choix de la représentation de l'environnement est crucial dans le processus de cartographie. Nous avons défini une représentation qui soit utilisable par le robot terrestre et qui peut être construite en utilisant les données provenant d'un drone, d'un robot au sol ou d'une base de données du réseau routier.

Nous avons choisi d'utiliser une représentation hybride, topologique et métrique. Ce choix est motivé par les capacités de hiérarchisation, de compacité et de modularité de cette représentation.

Le processus de création de cette carte utilise une orthoimage dans laquelle l'espace navigable est extrait grâce à différentes techniques. Cette classification permet de générer une image de traversabilité binaire dont le squelette filtré est calculé en utilisant l'algorithme delta medial axis. Le graphe est construit en parcourant ce squelette, les nœuds sont les intersections et les impasses, et les arêtes sont les chemins entre ces nœuds. Les informations métriques sont ajoutées grâce aux transformées en distance présentes dans le squelette. Certaines informations sémantiques sont également ajoutées lorsque elles sont disponibles et utiles.

Les cartes ainsi créées par les robots ont toutes la même structure et représentent le même environnement, mais sont perçues de points de vue très différents. Nous développons dans la suite des techniques de fusion appropriées.

Licence CC BY-NC-ND 3.0

3. Fusion de cartes aéro-terrestres

Sommaire

Introduction **62**
3.1. Appariement point à point **62**
3.2. Descripteur et mesure de dissimilarité **65**
3.2.1. Invariance, discriminance et complétude 65
3.2.2. Descripteur, structuration des informations utiles 66
3.2.3. Mesure de dissimilarité 67
3.3. Algorithmes de Matching Hiérarchique **70**
3.3.1. Algorithme Hongrois 70
3.3.2. Mesure ≪ raffinée ≫ 72
3.3.3. Suppression des assignations erronées 75
3.3.4. Matching topologique 76
3.3.5. Assignation par consensus de projection 77
3.3.6. Résultat du matching 78
3.3.7. Validation du matching 80
3.4. Analyse et discussion de l'approche proposée **81**
3.4.1. Analyse de résultats 81
3.4.2. Méthode d'optimisation alternative 83
3.4.3. Conclusion 84
3.5. Fusion des cartes **85**
3.5.1. Stratégie adaptée au contenu des cartes 86
3.5.2. Carte augmentée construite par projection 86
3.6. Utilisation de la carte fusionnée **88**
Conclusion **90**

Introduction

La coopération aéro-terrestre appliquée à la cartographie de l'espace navigable terrestre a pour objectif de fournir au robot terrestre une carte cohérente et utilisable pour la navigation et la planification. Pour cela, nous avons choisi une représentation hybride qui se concentre sur les points clés de l'environnement terrestre. Nous venons de présenter les moyens utilisés pour construire ce type de carte à partir des points de vue aérien et terrestre. Le but est maintenant de « fusionner » ces cartes qui ont des caractéristiques de précision et d'étendue différentes.

La fusion des cartes se fait en deux étapes :

1. On cherche à déterminer la position relative des cartes. Pour cela, il faut détecter les zones communes et calculer la transformation affine entre les cartes. Cela permet de superposer les deux cartes.
2. On crée ensuite une carte globale en fusionnant les deux cartes.

Dans notre cas, les zones communes doivent être détectées sans information apriori et la superposition des cartes ne peut pas se faire à l'aide d'une transformation affine à cause des déformations de la carte terrestre. La fusion est faite en gardant les informations les plus précises de chacune des deux cartes.

Le processus qui permet de détecter et fusionner des parties communes entre deux cartes géographiques est appelé « *map conflation* ». Dans notre contexte, nous nous plaçons dans le sous-domaine particulier qui étudie l'association de cartes routières (*Road Network Matching*). Il implique un processus d'identification des points situés sur différentes cartes au moyen d'informations géométriques, topologiques ou sémantiques. Ce processus est appelé « *matching* ». Ce terme peut être traduit en français par « mise en correspondance » ou par « appariement » dans le contexte de la théorie des graphes. Il consiste à associer des éléments entre eux grâce à un critère particulier.

Dans ce chapitre, nous présenterons notre approche de la fusion de cartes aéro-terrestres. Nous commencerons par présenter la méthode classique utilisée pour le matching de graphe. Nous définirons ensuite une mesure de dissimilarité entre nœuds adaptée à notre contexte. Par la suite, nous détaillerons toutes les étapes de l'algorithme de matching hiérarchique et discuterons les résultats. Finalement, nous expliquerons la stratégie de fusion adaptée au contenu des cartes.

3.1. Appariement point à point

En robotique, plusieurs solutions au problème de fusion de cartes ont déjà été proposées. Cependant, elles ont souvent des hypothèses fortes : environnements intérieurs, robots homogènes ou cartes sans déformations [Ferrao et al., 2017]. Les méthodes de matching point à point utilisent la topologie des environnements pour

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

déterminer les zones communes. Pour cela, la structure des graphes est utilisée et les méthodes dérivent des algorithmes de *graph matching*. Un graphe peut également être extrait de cartes métriques en utilisant le graphe de poses ou en calculant le diagramme de Voronoï d'une grille d'occupation [Bonanni et al., 2014].

Il existe plusieurs types de matching de graphes en fonction de leurs caractéristiques. Lorsque les nœuds ou les arêtes des cartes contiennent des informations métriques ou sémantiques, le problème d'appariement est nommé « *attributed graph matching* » (AGM) [Athanasios Mademlis, 2010]. Lorsque les éléments du graphe sont identifiés (ils portent un index), et ne sont donc pas interchangeables, le problème est appelé « *labeled graph matching* » [Abdulkader, 1998]. Dans tous les cas, le but est d'associer les nœuds des graphes entre eux en utilisant les informations qu'ils portent. Comme souvent, on sera amené à considérer la mise en correspondance de sous-parties de graphes différents. Le problème est donc considéré comme « *subgraph matching* » (SGM) ou « sous-graphe commun maximum » (MCS). Le matching de graphes permet de comparer et de reconnaitre les objets représentés par des graphes, il est donc naturellement utilisé dans beaucoup de domaines différents tels que le traitement d'image pour faire la reconnaissance de forme [Sebastian and Kimia, 2001], la chimie moléculaire, ou encore la médecine [Almasi et al., 2018].

La figure 3.1 illustre le problème de mise en correspondance de graphes. Il faut trouver les associations de nœuds (traits orange) qui permettent d'identifier les parties communes entre deux graphes.

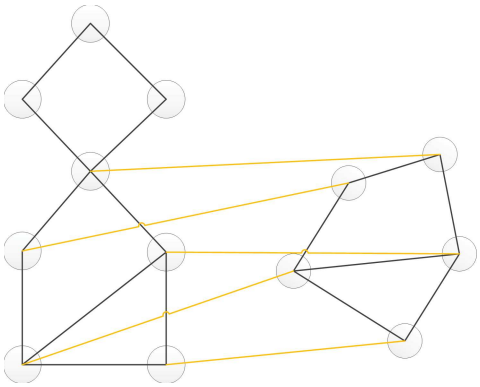


Figure 3.1. – Présentation du matching de nœuds.

Le problème d'appariement point à point de deux graphes G_1 et G_2 consiste à associer les nœuds et les arêtes du graphe G_1 à ceux du graphe G_2 au regard d'un cout d'association C.

Considérons $G_1 = \{N_1, E_1\}$ et $G_2 = \{N_2, E_2\}$, le problème peut être exprimé sous la forme d'un problème d'optimisation appelé « problème d'affectation quadratique » (QAP) [Heffley, 1972] :

$$P = \underset{P \in Per(|N_1|,|N_2|)}{\operatorname{argmin}} \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} c_{ij}p_{ij} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{i'=1}^{N_1}\sum_{j'=1}^{N_2} c_{iji'j'}p_{ij}p_{i'j'} \tag{3.1}$$

— $p_{ij} \in \{0, 1\}$ est l'association du nœud $n_i \in N_1$ et $n_j \in N_2$, la matrice d'assignation P appartient à l'ensemble des matrices de permutation de taille $|N_1| \times |N_2|$.
— c_{ij} est le cout d'assignation des nœuds n_i et n_j.
— $c_{iji'j'}$ est le cout d'assignation des arêtes $(n_i, n_{i'}) \in E_1$et $(n_j, n_{j'}) \in E_2$.

De part sa nature quadratique et combinatoire, la complexité de ce problème a été prouvé NP-difficile [Heffley, 1972]. Il n'existe pas d'algorithme pour résoudre ce problème de façon optimale en temps polynomial. Ainsi, les recherches se sont concentrées sur l'obtention de solutions sous-optimales en relâchant les contraintes du problème. C'est le cas de l'algorithme *Path following* [Zaslavskiy et al., 2009]. D'autres approches ont été envisagées comme l'utilisation de propriétés spectrales [Leordeanu and Hebert, 2005], de la distance d'édition de graphe [Luan, 2012] ou encore d'algorithmes génétiques [Montek Singh et al., 1997].

La solution que nous avons utilisée est de réduire la complexité du problème. Pour cela, il suffit de résoudre la partie linéaire du problème (Équation 3.1) en incluant les informations des arêtes dans les descripteurs des nœuds [Almasi et al., 2018]. Le problème est alors réécrit comme un problème d'affectation linéaire (Équation 3.2) qui peut être résolu grâce à un algorithme optimal bien connu.

$$P = \underset{P \in Per(|N_1|,|N_2|)}{\operatorname{argmin}} \; tr(C^T P) \tag{3.2}$$

La matrice de couts C est construite grâce aux couts d'assignation pour chaque paire de nœuds possible $c_{ij} = D(n_i, n_j) \,\forall\, (n_i, n_j) \in (N_1, N_2)$. $D(n_i, n_j)$ est une fonction de dissimilarité entre deux nœuds, elle définit le cout d'assignation. Cette matrice de couts peut être représentée sous la forme d'une grille où l'intensité dépend de la similitude entre les nœuds (Figure 3.2).

L'élément crucial de ce matching est la définition d'une mesure de dissimilarité qui représente la différence entre les nœuds $D(n_i, n_j)$. Ensuite, il faut résoudre le problème d'optimisation (Équation 3.2) qui permet d'associer les nœuds au regard du cout total d'assignation. Dans la plupart des cas, il existe plusieurs nœuds similaires et la solution n'est pas évidente.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Figure 3.2. – Représentation du problème sous forme de carte de couts. La matrice de couts C est représentée sous la forme d'une grille où l'intensité dépend de la similitude entre les nœuds. Les lignes sont les nœuds de la carte terrestre et les colonnes sont les nœuds de la carte aérienne.

3.2. Descripteur et mesure de dissimilarité

La mesure de dissimilarité n'est pas appliquée directement sur les nœuds car ils ne possèdent pas les mêmes informations selon leur source. Il est nécessaire de construire au préalable un *descripteur* de nœuds qui possède les mêmes informations et structures quelle que soit la carte source afin de pouvoir les comparer. Cette approche est similaire à celle utilisée en reconnaissance d'images. Le cout d'assignation des nœuds est pertinent s'il respecte les critères d'invariance et de discriminance.

3.2.1. Invariance, discriminance et complétude

L'invariance sous-entend que le cout entre deux nœuds représentant le même lieu géographique $\mathcal{L}$ est nul quelles que soient les cartes dont ils proviennent. Soit $\mathcal{L}_n$ le lieu géographique représenté par le nœud n :

$$\mathcal{L}_{n1} = \mathcal{L}_{n2} \implies D(n_1, n_2) = 0$$

La discriminance signifie que le cout entre des nœuds représentant des lieux géographique différents est non nul :

$$\mathcal{L}_{n1} \neq \mathcal{L}_{n2} \implies D(n_1, n_2) > 0$$

Si la fonction $D(n_i, n_j)$ est à la fois invariante et discriminante, elle est complète et respecte la forme suivante :

$$\mathcal{L}_{n1} = \mathcal{L}_{n2} \iff D(n_1, n_2) = 0$$

Licence CC BY-NC-ND 3.0

3.2.2. Descripteur, structuration des informations utiles

Les cartes peuvent avoir une source, une échelle et un repère différents. De plus, les déformations de la carte terrestre empêchent le calcul d'une transformation rigide entre les deux cartes. Dans l'absence d'un référentiel absolu commun comme le GPS, les informations de position des nœuds ne sont donc pas utilisables pour associer les nœuds. Les informations discriminantes sont alors les informations à propos de la « forme » du nœud :

- Le degré du nœud (le nombre d'arêtes auxquelles il est connecté).
- Le rayon (la transformée en distance au point de l'intersection).
- La forme des arêtes incidentes (une liste de points définis dans le repère local du nœud).

Ces informations sont disponibles dans les cartes, qu'elles proviennent du robot terrestre ou du drone. Ainsi nous définissons le descripteur d'un nœud comme suit :

$$Descripteur = \begin{cases} Degré & d_n \\ Rayon & R_n \\ Angles\,d'incidence & \theta_a,\ a \in [1, d_n] \\ Chemins\,incidents & \mathcal{C}_a,\ a \in [1, d_n] \end{cases}$$

Les chemins $\mathcal{C}_a$ sont définis par une liste de points dont le repère est centré sur le nœud, comme dans la « Donnée » portée par les arêtes. L'orientation du référentiel est l'orientation du robot à l'instant 0. Ce descripteur est illustré dans l'image suivante (Figure 3.3). Le rayon est représenté par un cercle vert, les chemins et angles d'incidences sont représentés avec une couleur différente pour chaque branche $a \in [1, d_n]$.

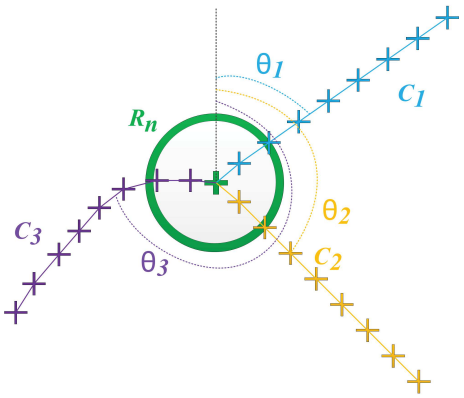


Figure 3.3. – Illustration du descripteur d'un nœud.

Ce descripteur est ensuite utilisé dans la mesure de dissimilarité qui décrit la différence entre deux nœuds.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

3.2.3. Mesure de dissimilarité

Cette mesure de dissimilarité doit être invariante à tout ce qui peut changer lors de la représentation d'un même lieu géographique sur des cartes différentes (Figure 3.4) : la translation, l'orientation, l'échelle mais également la limite du champ de vision du robot terrestre.

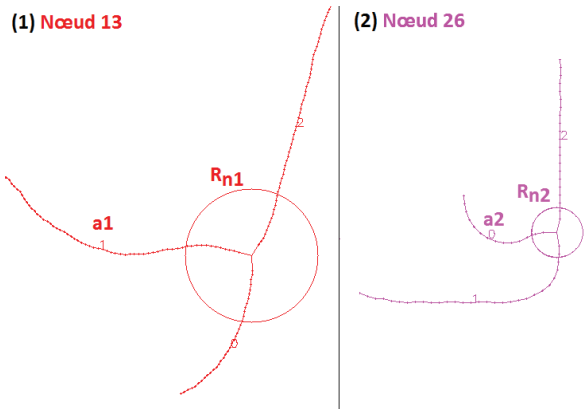


Figure 3.4. – Deux représentations différentes d'un même lieu géographique. (1) Perception terrestre, (2) perception aérienne.

Nous proposons la mesure de dissimilarité suivante :

$$D(n_1, n_2) = \begin{array}{l} K_d \, |d_{n1} - d_{n2}| + \\ \min\limits_{\alpha \in [0,2\pi]} \left[K_c \, Eq_\alpha(n1, n2) + K_a \sum\limits_{a1 \in [1, d_{n1}]} \min\limits_{a2 \in [1, d_{n2}]} (\theta_{a1} - \theta_{a2} + \alpha) \right] \end{array} \tag{3.3}$$

Cette mesure est composée de deux termes : le premier est la différence entre les degrés topologiques des nœuds, et le deuxième exprime la différence de forme des nœuds. Cette dernière est obtenue en calculant le minimum sur l'angle α de l'erreur quadratique entre les branches des deux nœuds (Équation 3.4) et de l'erreur angulaire entre les arcs proches. Les coefficients K_d, K_c et K_a permettent de pondérer la mesure et de contrôler l'importance des différents éléments et seront discuté dans la partie expérimentale.

L'erreur quadratique entre les branches des nœuds est calculée pour tous les angles $\alpha \in [0, 2\pi]$ avec un pas fixe $\Delta\alpha$. Elle correspond à la somme des distances au carré entre les points des branches :

$$Eq_\alpha(n_1, n_2) = \sum_{a1 \in [1, d_{n1}]} \sum_{p_1 \in \mathcal{C}_{a1}} \left(\min_{p_2 \in \mathcal{C}_{a2}} \| \, p_1 - H_\alpha(p_2) \, \|^2 \right) \tag{3.4}$$

Les points p_1 et p_2 sont respectivement deux points sur les chemins $\mathcal{C}_{a1}$ et $\mathcal{C}_{a2}$. $H_\alpha(p2)$ est la transformation rigide qui translate n_2 sur n_1, réalise la rotation α et applique

Licence CC BY-NC-ND 3.0

le facteur d'échelle $\lambda = R_{n1}/R_{n2}$. Il en résulte la superposition des deux nœuds avec une mise à l'échelle commune. Cette superposition est visible dans l'image suivante (Figure 3.5) (le nœud n_1 est en rouge (a), et le nœud n_2 est en violet (b)). Leur rayon est utilisé pour la mise à l'échelle (en noir). Les nœuds sont superposés (c) puis l'erreur entre les branches est calculée sur les distances en orange. Dans cet exemple, la valeur minimale de dissimilarité est obtenue pour $\alpha = 14°$, figure 3.5 (d).

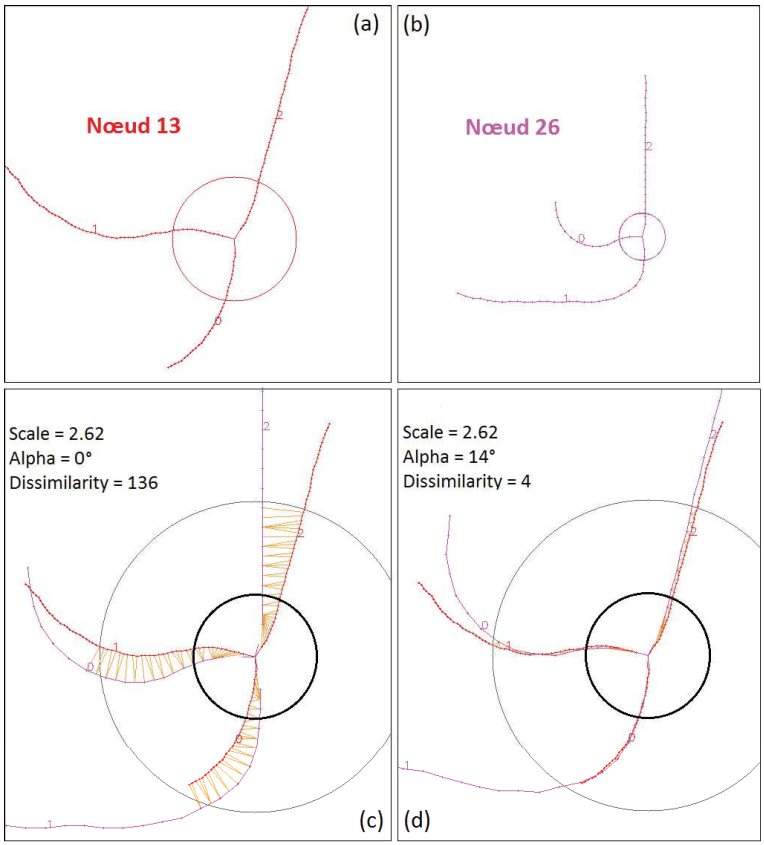


Figure 3.5. – Mesure de différence entre deux nœuds pour la même intersection. (a) Nœud du graphe 1, (b) Nœud du graphe 2, (c) Superposition des nœuds pour $\alpha = 0$, (d) Dissimilarité minimale obtenue pour $\alpha = 14°$.

Pour réduire le temps de calcul de l'erreur quadratique, les distances sont calculées uniquement pour les points appartenant aux branches les plus proches.

$$a2 = \underset{a2 \in [1, d_{n2}]}{argmin}(\theta_{a1} - \theta_{a2} + \alpha)$$

Cette mesure de dissimilarité $D(n_1, n_2)$ est finalement seuillée à une valeur C_{max} pour définir le cout d'assignation c_{n_1,n_2}.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Lors du calcul de la différence entre les deux descripteurs, l'orientation minimale ainsi que le facteur d'échelle sont sauvegardés. Ainsi, on peut définir les fonctions :

$$\alpha(n_1, n_2) = \underset{\alpha \in [0,2\pi]}{argmin} \left[\sum_{a1 \in [1, d_{n1}]} \sum_{p1 \in C_{a1}} (\min_{p2 \in \mathcal{C}_{a2}} \| p1 - H_\alpha(p2) \|^2) \right]$$

et $\lambda(n_1, n_2) = R_{n1}/R_{n2}$

Discussion :

La formulation de la mesure de dissimilarité (Équation 3.3) est, par construction, quasi invariante aux éléments de transformations :

- Échelle : les informations sur les nœuds sont normalisées par rapport aux rayons.
- Translation : la mesure ne prend pas en compte la position du nœud, les points des arêtes sont définis dans un repère centré sur le nœud.
- Orientation : la mesure est calculée pour toutes les orientations possibles, la plus faible est conservée.

Le critère d'invariance n'est pas complètement satisfait à cause des effets liés à l'échantillonnage. Que ce soit pour l'échantillonnage des points des arêtes ou de l'angle de rotation, ces variations peuvent entrainer un changement des valeurs de la mesure. On note cependant que, de manière générale, ces variations sont faibles.

La discriminance est élevée bien qu'elle puisse être mise en défaut par de l'aliasing perceptuel (deux lieux différents ont le même descripteur) comme dans le cas de structures de type Manhattan, où tous les carrefours ont la même forme et la même dimension.

L'impact des pondérations (K_i) sur la mesure de dissimilarité et sur l'association des nœuds a été évalué et sera présenté dans le chapitre 4 (Expérimentations). Le choix a été fait d'équilibrer l'impact de chaque terme. Les valeurs utilisées dans tous les scénarios expérimentaux sont présentées dans le tableau suivant (Tableau 3.1).

K_d	K_c	K_a	C_{max}	$\Delta\alpha$
20	$\frac{1}{20}$	$\frac{1}{5}$	100	2°

Table 3.1. – Liste des paramètres utilisés.

Des améliorations locales ont été utilisées pour rendre cette mesure de dissimilarité plus robuste. L'échantillonnage des chemins peut varier entre les cartes terrestre et aérienne, et le calcul de l'erreur quadratique peut en être affecté. Une interpolation pour équilibrer le nombre de points dans les chemins réajuste ce changement de représentation. Les chemins n'ayant pas été explorés ne sont représentés qu'avec

Licence CC BY-NC-ND 3.0

la perception locale du robot terrestre, et souffrent donc des effets de bords dûs à la bird-eye view. Pour réduire l'impact de ce phénomène, le calcul de l'erreur quadratique est limité aux points proches du nœud ($\overline{\mathcal{C}_{a1}} = \{p_1 \mid \|p_1\| \leq 2R_{n1}$).

Dans d'autres contextes moins contraints, d'autres mesures de dissimilarité peuvent être utilisées. Lorsque les cartes possèdent le même référentiel, une mesure utilisant la position des nœuds peut être utilisée, ce qui augmente la discriminance de la mesure [Hackeloeer et al., 2013]. Des mesures n'utilisant que les angles d'incidence et/ou le degré des nœuds peuvent être utilisées malgré leur faible discriminance si elles sont accompagnées d'un procédé de matching complexe [Huang and Beevers, 2005]. Certaines solutions provenant du traitement d'image, comme les descripteurs SIFT, peuvent être utilisées sur des représentations utilisant des grilles d'occupation, c'est le cas des travaux de Ferrao [Ferrao et al., 2017].

3.3. Algorithmes de Matching Hiérarchique

La fonction de dissimilarité entre deux descripteurs de nœuds permet de calculer un cout d'assignation entre ces nœuds. On peut ainsi construire la matrice C qui contient les couts d'assignations entre tous les couples de nœuds possibles $C = \{c_{ij}\}$, $c_{ij} = D(n_i, n_j) \forall (n_i, n_j) \in (N_1, N_2)$. Le problème linéaire d'assignation (Équation 3.2) est résolu grâce à un algorithme optimal appelé méthode Hongroise.

3.3.1. Algorithme Hongrois

L'algorithme Hongrois, aussi appelé algorithme de Kuhn-Munkres [Kuhn, 1955], est un algorithme d'optimisation combinatoire qui résout le problème d'affectation en temps polynomial ($\mathcal{O}(n^4)$, n étant le nombre de nœuds) et de manière optimale. Il permet de trouver une association de poids minimum entre les nœuds. Le problème d'assignation devient donc :

$$P = \min_{P \in Per(N_{max})} tr(C^T P),\ N_{max} = \max(|N_1|, |N_2|)$$

Afin d'appliquer cet algorithme pour l'association de graphes de tailles différentes, nous ajoutons des nœuds virtuels au graphe le plus petit. La valeur d'assignation de ces nouveaux éléments est maximal (C_{max}). De cette façon, les nœuds en surplus dans l'un des graphes ne seront pas assignés.

La méthode Hongroise permet de sélectionner un seul élément par ligne et par colonne ($P \in Per(N_{max})$) en minimisant la somme des couts ($tr(C^T P)$). L'algorithme cherche à créer des *zéros* dans les lignes et les colonnes en soustrayant l'élément le plus petit. Il renvoie la liste des associations $\mathcal{A} = \{a = (i, j) \mid P_{i,j} = 1\}$. On définit alors la fonction $\mathcal{A}(n_i) = n_j$ qui renvoie le nœud associé au nœud donné en

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

paramètre. Le pseudo-code montrant le déroulement de la méthode Hongroise est présenté ci-dessous.

Algorithm 3.1 Algorithme Hongrois

Entrées : C matrice carrée
Sorties : $\mathcal{A}$ liste des assignations

Initialisation : $\mathcal{A} \leftarrow$ un ensemble indépendant de zéros de taille maximale dans C
tant que $|\mathcal{A}| < |N_{max}|$ **faire**
  pour Chaque élément (i,j) de $\mathcal{A}$ **faire**
    Marquer la ligne i et la colonne j
    $\omega \leftarrow$la plus petite valeur non marquée dans C
    pour i' allant de 1 à $|N_{max}|$ **faire**
      pour j' allant de 1 à $|N_{max}|$ **faire**
        si $c_{i',j'}$ n'est pas marqué **alors**
          $c_{i',j'} = c_{i',j'} - \omega$
        fin si
        si $c_{i',j'}$ est marqué deux fois **alors**
          $c_{i',j'} = c_{i',j'} + \omega$
        fin si
      fin pour
    fin pour
  fin pour
fin tant que
Return $\mathcal{A}$

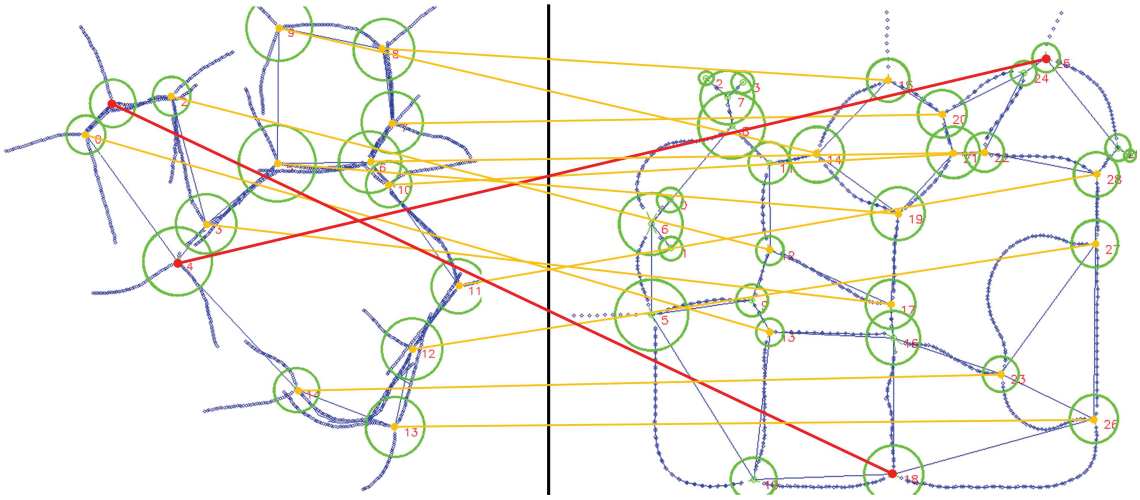


Figure 3.6. – Résultat de l'algorithme Hongrois utilisant la mesure initiale. Les traits rouges indiquent les mauvaises associations.

Le résultat du matching en utilisant la méthode Hongroise sur la matrice de couts construite avec la mesure de dissimilarité précédente est présenté en figure 3.6.

Licence CC BY-NC-ND 3.0

Les cartes utilisées sont celles présentées dans le chapitre précédent. Les traits représentent les associations de nœuds. Les traits rouges montrent les erreurs d'association. Ces erreurs d'association proviennent des similarités entre les nœuds. Le cout d'association entre ces éléments est donc faible. De plus, lorsqu'une route n'a pas été explorée par le robot terrestre, sa représentation est construite à partir de la perception locale dans la bird-eye view qui est limitée dans l'espace. Les tests effectués dans des environnements simulés et réels ont montré la pertinence de la fonction cout et de l'algorithme Hongrois.

La représentation du matching grâce à une carte de couts permet de voir les nœuds assignés et leur couts respectifs.

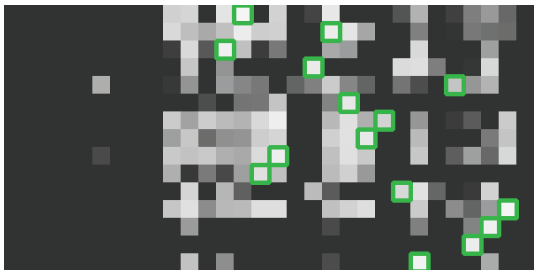


Figure 3.7. – Carte de couts assignée avec la méthode Hongroise. La matrice de couts C, est représentée sous forme de grille où l'intensité dépend de la similitude entre les nœuds. Les lignes correspondent aux 15 nœuds de la carte terrestre et les colonnes sont les 30 nœuds de la carte aérienne. La solution renvoyée par l'algorithme Hongrois est présentée en vert.

Ce premier matching permet d'avoir une approximation initiale du positionnement relatif des cartes. Cette information peut alors être utilisée pour raffiner la mesure de dissimilarité.

3.3.2. Mesure « raffinée »

En faisant l'hypothèse qu'une majorité des associations données par l'algorithme précédent est correcte, on peut faire une analyse statistique des mesures précédentes. Cela permet d'estimer la transformation moyenne pour passer d'une carte à l'autre, ainsi que l'écart-type. Pour réduire le nombre d'associations erronées dans ce calcul, nous n'utilisons que les associations ayant un cout strictement inférieur à C_{max}, soit :

$$\mathcal{A}^\star = \{a \in \mathcal{A} \mid c(a) < C_{max}\}$$

On estime les valeurs moyennes et les écarts-types des angles de rotation et du facteur d'échelle entre les deux cartes :

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

$$\overline{\alpha} = \frac{1}{|\mathcal{A}^\star|}\sum_{a\in\mathcal{A}^\star}\alpha(a),\ \sigma_\alpha = \sqrt{\frac{1}{|\mathcal{A}^\star|}\sum_{a\in\mathcal{A}^\star}\|\alpha(a)-\overline{\alpha}\|^2}$$

$$\overline{\lambda} = \frac{1}{|\mathcal{A}^\star|}\sum_{a\in\mathcal{A}^\star}\lambda(a),\ \sigma_\lambda = \sqrt{\frac{1}{|\mathcal{A}^\star|}\sum_{a\in\mathcal{A}^\star}\left\|\lambda(a)-\overline{\lambda}\right\|^2}$$

Une transformation rigide globale ne peut pas être calculée à cause des déformations de la carte terrestre. Néanmoins, en utilisant ces informations, on peut créer une nouvelle mesure de dissimilarité qui prend en compte l'orientation et la taille des nœuds :

$$D_2(n_1,n_2) = \begin{array}{l} K_d\left|d_{n1}-d_{n2}\right| + K_\lambda(\sigma_\lambda)\left|\overline{\lambda}-\lambda\right| + K_\alpha(\sigma_\alpha)\left|\overline{\alpha}-\alpha\right| + \\ \min\limits_{\alpha\in[\overline{\alpha}\pm 3\sigma_\alpha]}\left[K_c Eq_\alpha(n1,n2) + K_a\sum\limits_{a1\in[1,d_{n1}]}\min\limits_{a2\in[1,d_{n2}]}(\theta_{a1}-\theta_{a2}+\alpha)\right] \end{array} \tag{3.5}$$

Cette mesure permet d'augmenter la discriminance de la mesure initiale. En réduisant la plage de recherche de l'angle α, le risque de symétrie par rotation est réduit. Les coefficients de pondération K_λ et K_α peuvent dépendre des écarts-types. Plus l'écart-type est faible, plus le coefficient serait élevé. Cette mesure de différence raffinée est également seuillée à la valeur C_{max} pour définir le nouveau cout d'assignation.

On peut ainsi construire une nouvelle matrice de couts C_2. Pour gagner du temps de calcul, seuls les couples de nœuds avec un score $D(n_1,n_2) < C_{max}$ sont mis à jour. En effet, dans la pratique, cette mesure de différence ne peut être que supérieure à la mesure précédente. On applique de nouveau la méthode Hongroise pour déterminer le matching raffiné. La figure 3.8 montre le résultat du matching en utilisant cette mesure raffinée. L'association en rouge est erronée et le point rouge montre un nœud non matché.

Licence CC BY-NC-ND 3.0

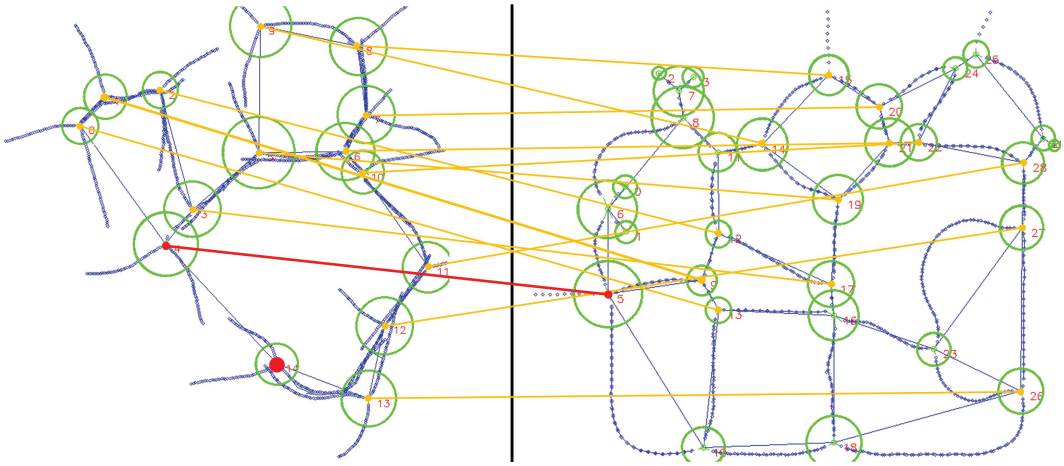


Figure 3.8. – Résultat de l'algorithme Hongrois utilisant la mesure raffinée. Les éléments en rouge sont mal assignés.

L'utilité de la mesure raffinée est clairement visible sur les cartes de couts (Figure 3.9). Le nombre de candidats pour l'assignation de chaque nœud est grandement réduit.

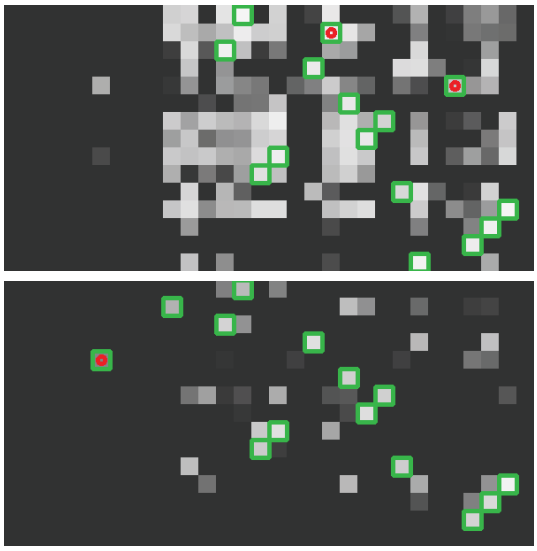


Figure 3.9. – (haut) Matching avec la mesure de dissimilarité initiale, (bas) Matching avec la mesure de dissimilarité raffinée. Les assignations erronées sont marquées d'un point rouge.

Le résultat de ce nouveau matching est meilleur que le précédent puisqu'il associe correctement un plus grand nombre de nœuds. Cependant, il n'est toujours pas parfait au regard des nœuds qui ne sont toujours pas associés. De plus, des matchings erronés peuvent être encore présents. Pour rendre le processus plus robuste, nous proposons, dans les sections suivantes, une approche hiérarchique d'appariement et de détection des assignations erronées.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

3.3.3. Suppression des assignations erronées

Le matching de nœuds définit implicitement le matching des arêtes. En effet, s'il existe une arête $e_1 \in E_1$ entre deux nœuds du graphe G_1 associés à deux nœuds du graphe G_2 qui possèdent également une arête commune $a_2 \in E_2$, alors on peut considérer que les arêtes a_1 et a_2 sont associées entre elles. Cette notion est présentée dans la prochaine illustration (Figure 3.10). Les associations de nœuds sont les arcs et les associations implicites des arêtes sont les doubles flèches bleues.

Grâce à cette assignation implicite des arêtes, il est possible de détecter des nœuds mal matchés. En effet, si aucun nœud adjacent (aussi appelé nœud voisin) à un nœud matché n'est associé aux nœuds voisins de son correspondant, alors ce nœud est mal matché. Cette procédure est formalisée comme suit. Soit $\delta(n)$ l'ensemble des voisins d'un nœud n, $\delta(n) = \{n_\delta \in N \mid \exists(n_\delta, n) \in E\}$:

$$\text{Si } a = (n_1, n_2) \in \mathcal{A} \ , \ \forall n_\delta \in \delta(n_1), \mathcal{A}(n_\delta) \notin \delta(n_2) \Longrightarrow \text{Enlever } a \text{ de } \mathcal{A}$$

L'illustration montre un exemple d'assignation erronée détectée grâce à cette méthode. Les nœuds associés par l'arc rouge ne possèdent pas de voisins associés entre eux, cette association est donc considérée comme erronée.

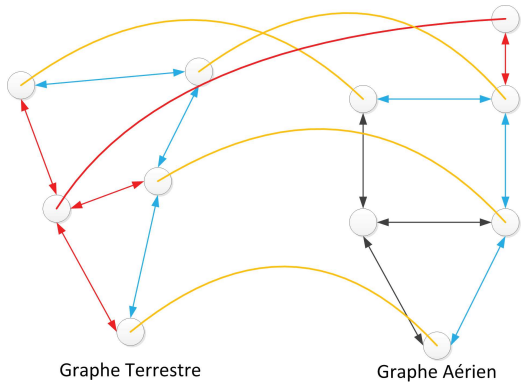


Figure 3.10. – Association implicite des arêtes et détection de faux matching. Les arcs montrent les associations entre les nœuds. Celles en jaune produisent l'association des arêtes en bleu. L'association en rouge est erronée car les arêtes rouges ne correspondent pas.

Licence CC BY-NC-ND 3.0

3.3.4. Matching topologique

Notons que cette approche peut également être étendue pour associer des nœuds entre eux. En effet, s'il existe un couple de nœuds n_1 et n_2 tel que leur association permet d'associer leurs arêtes incidentes alors cette assignation est ajoutée.

$$\text{Si } \exists! n_2 \mid \forall n_\delta \in \delta(n_1), \mathcal{A}(n_\delta) \in \delta(n_2) \Longrightarrow \text{On ajoute } (n_1, n_2) \text{ à } \mathcal{A}$$

Autrement dit, s'il existe un nœud non matché dont l'assignation permettrait d'associer toutes les arêtes qui lui sont connectées, alors on ajoute cette assignation à $\mathcal{A}$.

L'illustration suivante (Figure 3.11) montre un exemple d'associations obtenues en utilisant cette méthode. Les arcs jaunes représentent les associations initiales. Les arcs bleus et violets représentent le matching topologique grâce aux voisins, les arêtes correspondantes sont représentées de la même couleur.

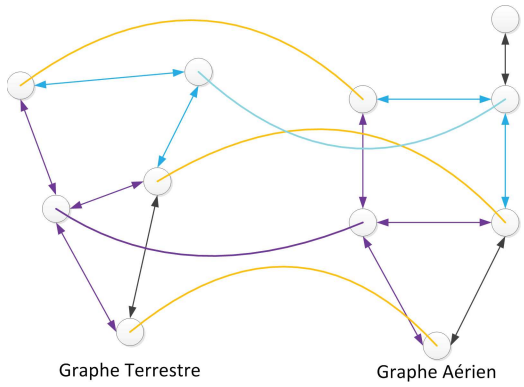


Figure 3.11. – Association de nœuds en utilisant les voisins matchés. Les arcs jaunes représentent les associations initiales. Les arcs bleus et violets représentent le matching topologique. Les arêtes correspondantes sont représentées de la même couleur.

Cette méthode est appliquée à tous les nœuds non matchés. Ces nœuds sont ordonnés par le nombre de leur voisins matchés. À chaque nouvelle association, cet ordre est recalculé pour profiter de la réaction en chaine. L'évolution de la carte de couts de l'exemple de matching en appliquant cette méthode est présentée en Figure 3.12. L'assignation supprimée est en rouge et celle qui est ajoutée est en bleu. On note qu'il reste un nœud non matché.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

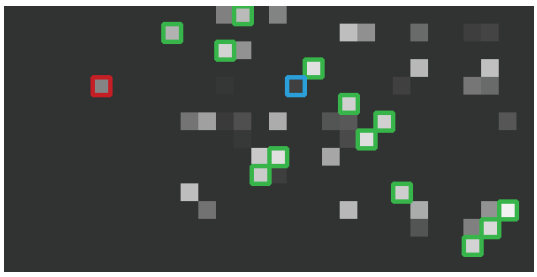


Figure 3.12. – Résultat du matching topologique. Matching initial en vert. En rouge l'assignation supprimée et en bleu celle ajoutée.

Cet algorithme ne peut pas résoudre le problème s'il existe plusieurs associations possibles (si n_2 n'est pas unique). Cela peut arriver lorsqu'il y a une symétrie topologique. Dans ce cas, aucune association n'est effectuée et une autre méthode doit être utilisée.

3.3.5. Assignation par consensus de projection

Dans la plupart des situations, la carte construite par le robot terrestre est déformée par la dérive de l'odométrie. Il n'est donc pas possible de calculer une transformation rigide globale adaptée à toute la carte. Cependant, les nœuds matchés nous permettent de calculer une transformation valide localement. En effet, on peut calculer la transformation locale H_{n1} à partir des positions de n_1 et $n_2 = \mathcal{A}(n_1)$, de l'angle de rotation $\alpha(n_1, n_2)$ et du facteur d'échelle $\lambda(n_1, n_2)$. Cette transformation n'est valide que si les nœuds sont correctement matchés.

On peut alors utiliser les projections locales des voisins matchés d'un nœud non matché n_1 pour trouver un nœud n_2 à lui associer. Si toutes les projections de n_1 sont proches du même nœud n_2 non matché alors, les nœuds n_1 et n_2 correspondent.

$$\text{Si } \exists n_2 \mid \forall n_\delta \in \delta(n_1), H_{n_\delta}(n_1) \approx n_2 \Longrightarrow \text{On ajoute } (n_1, n_2) \text{ à } \mathcal{A}$$

Les deux approches précédentes peuvent associer entre eux des nœuds qui ont un cout d'assignation élevé. Cela permet d'assigner des nœuds où les perceptions aériennes et terrestres sont trop différentes pour générer des descripteurs similaires. Dans la carte de couts (Figure 3.13), la nouvelle assignation en bleu est faite entre des nœuds ayant un cout d'assignation maximal.

Licence CC BY-NC-ND 3.0

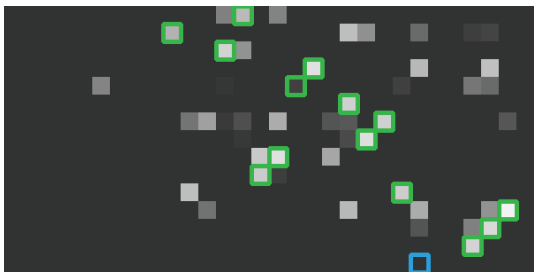


Figure 3.13. – Carte de couts après matching par projection. L'assignation précédente est en vert et la nouvelle assignation est en bleu.

3.3.6. Résultat du matching

En utilisant successivement les techniques précédentes, on obtient un matching de graphe cohérent. Le processus hiérarchique est schématisé ci-dessous (Figure 3.14).

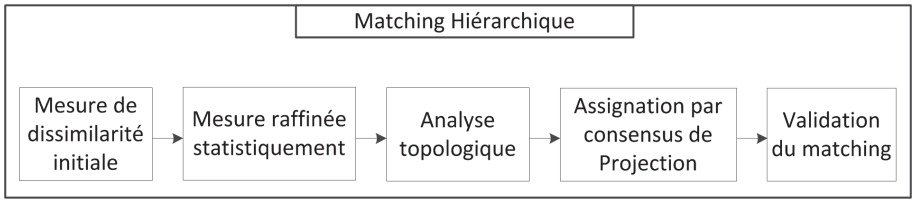


Figure 3.14. – Architecture du matching hiérarchique.

L'image suivante (Figure 3.15) montre le résultat final de l'exemple précédent après application du matching hiérarchique. Le trait rouge correspond au matching par projection, le trait bleu clair est l'association détectée grâce aux voisins en violet.

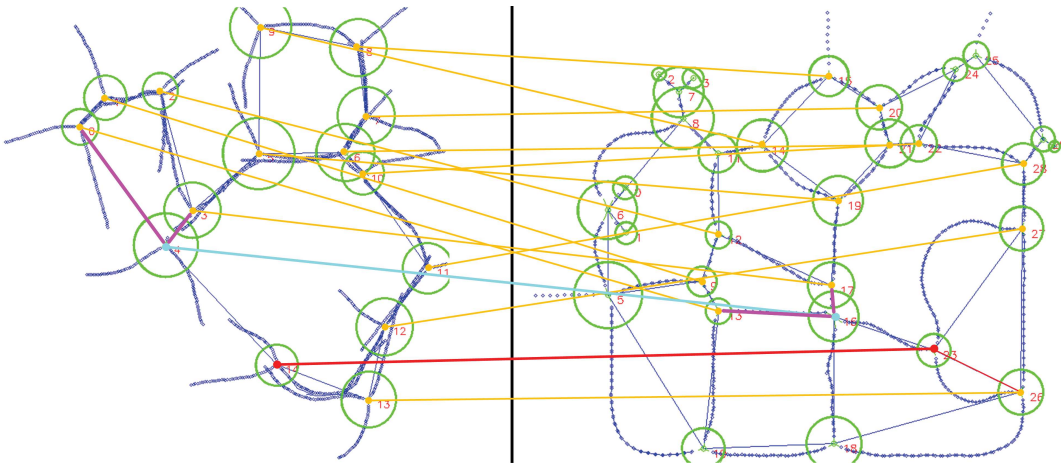


Figure 3.15. – Résultat du matching hiérarchique aéro-terrestre. Les traits jaunes correspondent au matching raffiné. Le trait rouge est l'assignation par projection, le trait bleu clair est l'association détectée grâce aux voisins en violet.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

L'algorithme complet pour la mise en correspondance des graphes aérien et terrestre est présenté dans le logigramme suivant (Figure 3.16) :

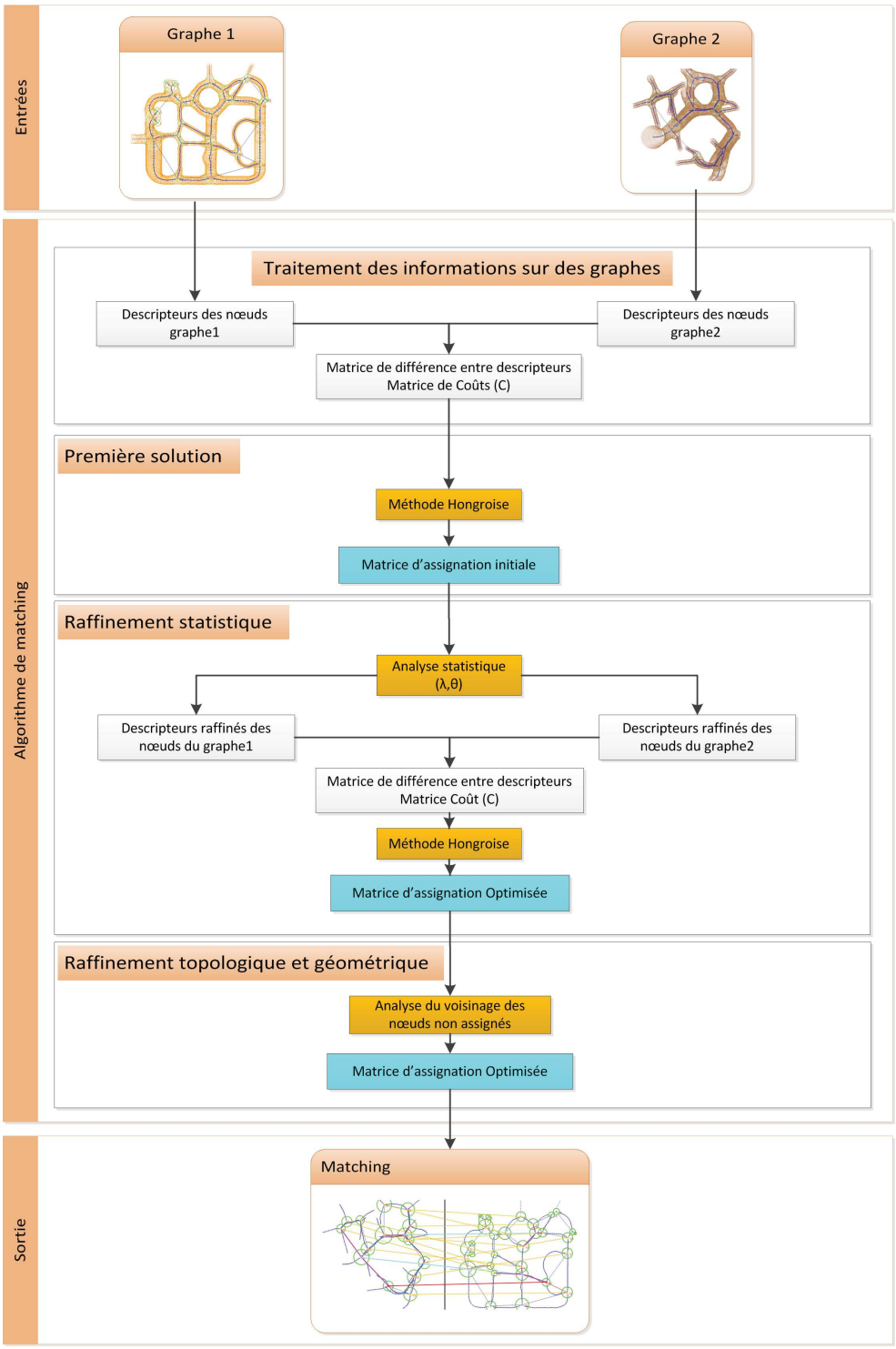


Figure 3.16. – Logigramme de l'algorithme de matching de graphes.

Licence CC BY-NC-ND 3.0

Dans l'exemple présenté tout au long de cette section, le matching est concluant. Il n'y a plus d'association erronée et tous les nœuds du plus petit graphe sont matchés. Ce matching peut donc être considéré comme bon. Il faut pouvoir évaluer la qualité du matching entre deux graphes de manière quantitative pour pouvoir fusionner sans risque les cartes. Il n'est cependant pas possible de déterminer à coup sûr si un nœud est correctement matché ou non.

3.3.7. Validation du matching

Il n'est pas évident de déterminer la qualité d'un matching sans information sur la taille, la complexité et la ressemblance des éléments présents dans un environnement. En dehors de ces informations, qui dépendent du contexte applicatif, plusieurs critères peuvent être pris en compte pour déterminer la qualité d'un matching de graphes [Huang and Beevers, 2005] :

— Le nombre de nœuds matchés ou non
— Le cout d'assignation des nœuds
— Le nombre d'arêtes matchées ou non

Le taux réel de chevauchement n'étant pas connu, un matching peut être correct même avec un nombre élevé d'arêtes et de nœuds non matchés.

En utilisant la formulation matricielle du problème ainsi que les matrices d'adjacence A_{G1} et A_{G2} des graphes, nous avons choisi de définir un score de matching comme suit :

$$\text{Score} = (1-\gamma)\frac{E_{min} - \left\|A_{G1} \times PA_{G2}P^T\right\|_F^2}{E_{min}} - \gamma\,\frac{tr(S_n^T P)}{N_{min}} \tag{3.6}$$

$E_{min} = \min(|E_1|, |E_2|)$ est le nombre maximum d'arêtes qu'il est possible d'associer. $\left\|A_{G1} \times PA_{G2}P^T\right\|_F^2$ est le nombre d'arêtes matchées calculé grâce à la norme de Frobenius appliquée au produit terme à terme entre A_{G1} la matrice d'adjacence du graphe G_1 et $PA_{G2}P^T$ la matrice d'adjacence du graphe G_2 projetée au regard de l'assignation P. Pour faciliter la comparaison avec d'autres algorithmes de matching, la matrice de couts C est transformée en matrice de similarité normalisée $S_n = \frac{C_{max}-C}{C_{max}}$. Ce second terme représente la similarité moyenne entre les nœuds matchés. $N_{min} = \min(|N_1|, |N_2|)$ est le nombre maximum de nœuds qu'il est possible d'associer.

Dans cette formulation, les nœuds non matchés sont pénalisants puisqu'ils ne rapportent pas de points de similarité. Le paramètre γ permet de pondérer l'importance de chacun des termes dans le score final. Le score est donc une valeur comprise entre $-\gamma$ et γ. Plus le score est faible, plus le matching est susceptible d'être correct. Une valeur de score égale à γ correspond au cas où toutes les associations sont mauvaises

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

(pas d'arêtes et cout élevé) alors que la valeur $-\gamma$ est le score parfait où toutes les arêtes sont matchées grâce à des associations de nœuds identiques. Pour l'exemple présenté dans ce chapitre, le score de matching est de -0.4 avec $\gamma = 0.5$.

La détection d'un matching correct se fait grâce à un seuil sur ce score de matching.

3.4. Analyse et discussion de l'approche proposée

Dans un premier temps, les résultats obtenus grâce à cette méthode seront décryptés. Ensuite, nous présenterons une méthode alternative utilisée pour comparer les résultats. Enfin, nous discuterons des avantages et des inconvénients de notre approche.

3.4.1. Analyse de résultats

Pour une première validation de l'approche, l'algorithme hiérarchique de matching a été testé sur un ensemble de 11 cartes terrestres et 13 cartes aériennes provenant de l'environnement Pavin. Les cartes représentent plusieurs niveaux de cartographie et plusieurs pourcentages de chevauchement. Pour obtenir une vérité terrain, les associations ont été vérifiées manuellement et chaque matching a été marqué comme correct ou non. Dès qu'un seul nœud est mal assigné, le matching est considéré comme « faux ». Lorsqu'aucun nœud n'est assigné, le matching est indiqué comme « non matché ». La taille d'un matching est le produit de la taille des graphes : $|N_1| \times |N_2|$, ce qui correspond à la taille de la matrice d'assignation. Le pourcentage de chevauchement est déterminé manuellement comme le nombre :

$$\text{chevauchement} = \frac{N_{commun}}{|N_1| + |N_2| - N_{commun}} \tag{3.7}$$

Le temps de calcul et le score des 143 matchings sont présentés dans les graphiques suivants : Figure 3.19 et Figure 3.17 respectivement.

Sur la figure 3.17, nous représentons le score de matching en fonction du pourcentage de chevauchement pour un paramètre $\gamma = 0.5$ et un seuil de validité fixé à -0.3. On peut voir que, à part quelques exceptions, le score (Équation 3.6) est un bon indicateur de la pertinence du matching. En effet, les points verts représentant les matchings corrects ont un score faible alors que les points rouges des « faux » matchings (ayant au moins une erreur) ont un score plus élevé. On peut remarquer que l'algorithme de matching est plus efficace sur des taux de chevauchement supérieur à 30%. Pour des taux plus faibles, et un nombre de nœuds faible, le score est moins informatif. Les points verts au-dessus du seuil correspondent à des matchings avec une initialisation faible. Dans ces cas, le matching hiérarchique utilise des techniques d'associations par adjacence et par projection qui permettent de trouver un bon matching malgré un cout d'assignation élevé.

Licence CC BY-NC-ND 3.0

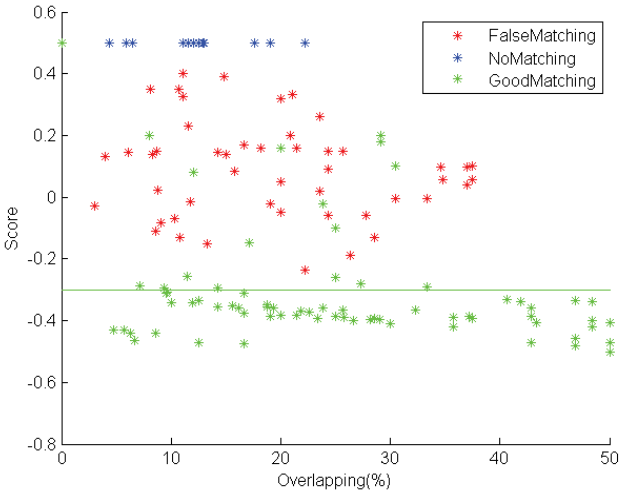


Figure 3.17. – Diagramme du score de matching.

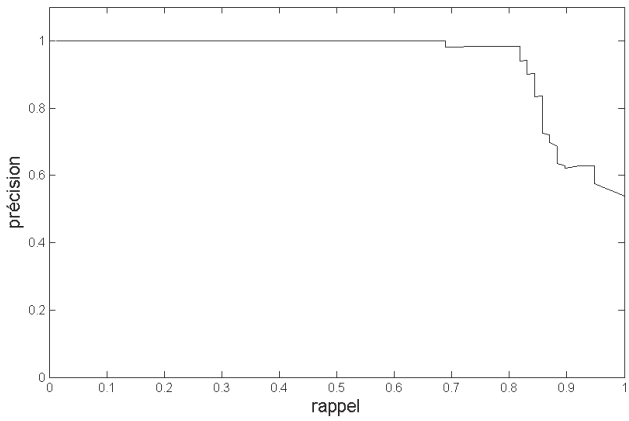


Figure 3.18. – Courbe de précision/rappel du score de matching.

La courbe de précision en fonction du rappel pour la score de matching est présentée en figure 3.18. Cette courbe montre la pertinence du score de matching. La valeur de précision reste proche de 1, même pour des valeurs de rappel élevées.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

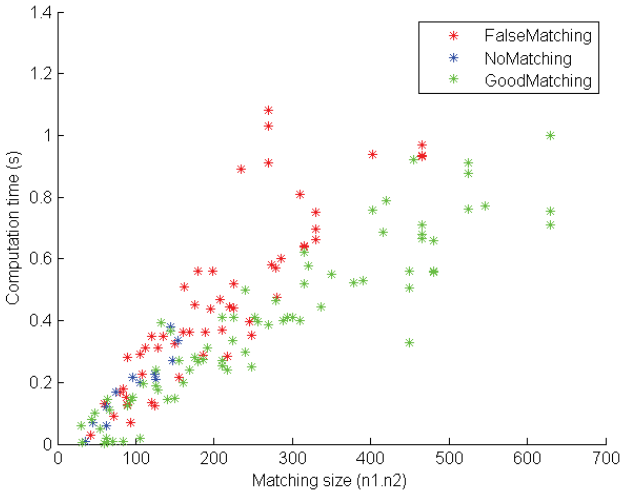


Figure 3.19. – Temps de calcul de l'algorithme.

Enfin, la figure 3.19 montre l'évolution du temps de calcul en fonction de la dimension des graphes ($|N_1| \times |N_2|$). On peut remarquer que le temps de calcul est plus court pour les matchings corrects. Cela est dû au fait que les techniques d'assignation topologique et par projection ne sont appliquées que sur les nœuds non matchés. Ainsi, les matchings qui ont une bonne initialisation, et donc moins de nœuds mal matchés, prennent moins de temps à être calculés. La plus grande partie du temps de calcul est prise par le calcul de la matrice couts. En effet, la mesure de dissimilarité entre les nœuds prend beaucoup de temps à cause du calcul pour toutes les orientations possibles. Une augmentation du pas de calcul $\Delta\alpha$ réduit le temps d'exécution mais ferait baisser la précision de la mesure.

3.4.2. Méthode d'optimisation alternative

Nous avons comparé notre méthode avec une autre solution qui consiste à adapter le score de matching et l'intégrer directement dans un problème d'optimisation. Le nouveau problème est formulé comme ceci :

$$\min_{P \in Per(|N_1|,|N_2|)} \left[(1-\gamma) \left\| A_{G1} - P A_{G2} P^T \right\|_F^2 + \gamma \, tr(C_n^T P) \right] \tag{3.8}$$

On retrouve le cout d'assignation et le nombre d'arêtes non matchées mais exprimés d'une autre manière. Ce problème est une formulation matricielle du problème d'assignation quadratique (Équation 3.1) mais sans le cout d'assignation entre arrêtes. Ce problème reste NP-difficile. Pour le résoudre, nous avons choisi d'utiliser l'algorithme *"Path Following"* proposé par Zaslavskiy [Zaslavskiy et al., 2009]. Il présente

Licence CC BY-NC-ND 3.0

de très bonnes performances au niveau de l'état de l'art actuel et est disponible au travers de la bibliothèque d'algorithmes « Graphm » [Graphm, 2009].

Les tests sur les graphes précédents ont montré que la résolution du problème quadratique prend 1000 fois plus de temps que la résolution du problème linéaire d'assignation. Cet algorithme permet une mise en correspondance des nœuds légèrement meilleure que le matching initial utilisant l'algorithme Hongrois. En effet, dans certain cas, il y a moins de nœuds mal assignés. Cependant, cette performance est obtenue pour une valeur de γ qui varie pour chaque matching. Lors de l'utilisation d'une valeur fixe, les résultats ne sont pas meilleurs que le matching en utilisant la méthode Hongroise. Ce manque de performance est dû au fait que la matrice de couts C n'est pas bien conditionnée.

Le temps de calcul étant beaucoup plus long et le résultat nécessitant, dans tous les cas, l'application des autres techniques de vérification, cette approche a été abandonnée.

3.4.3. Conclusion

L'approche proposée pour la mise en correspondance de cartes aériennes et terrestres s'inspire des méthodes de reconnaissance de graphes et de reconnaissance de formes. Elle cherche à identifier la meilleure assignation des nœuds de cartes partielles à partir de descripteurs et d'une mesure de dissimilarité adaptée. Cette association est effectuée de manière hiérarchique, en s'appuyant sur une mesure de dissimilarité entre les nœuds. La mesure de dissimilarité utilise les informations géométriques portées par ces derniers. Dans le cas présent, certaines informations ne sont pas accessibles apriori telles que le facteur d'échelle et le changement de référentiels entre les deux cartes. Ainsi, une première mesure de dissimilarité a été construite grâce aux informations de forme des nœuds (le nombre et la chemin des arêtes incidentes) pour être invariante à ces éléments inconnus. Ensuite, un matching avec ce cout d'assignation est déterminé en utilisant l'algorithme Hongrois. Ce matching initial permet de calculer le facteur d'échelle moyen ainsi que l'orientation moyenne pour une mesure de dissimilarité raffinée. Le matching est ensuite amélioré par la suppression d'associations erronées et l'ajout d'assignations grâce à la topologie et la projection locale cohérente du voisinage. Pour finir, le matching est validé ou invalidé grâce à un score construit avec le nombre d'arêtes matchées et la similitude entre les nœuds associés.

Cette approche est assez simple et modulaire dans sa structure. On peut facilement modifier les mesures de dissimilarité pour intégrer de nouveaux éléments comme la largeur des arêtes ou un cout de différence sémantique. Il est ainsi facile d'adapter cette approche à d'autres capteurs ou à d'autres sources d'informations, comme les cartes provenant de données en ligne. Les assignations par adjacence et par projection peuvent être alternées et utilisées sur les nœuds matchés pour vérifier la

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

cohérence du matching. Ce qui revient à ré-itérer le processus à l'image d'un RANSAC "topologique". Le score et le seuil de validité permettent de tester facilement si un matching est correct ou non.

Comme le montre la figure 3.19, le temps de calcul du matching peut être trop long pour être utilisé à une fréquence élevée.

Des différences importantes dans les représentations peuvent rendre le matching impossible. En effet, la représentation des intersections dépend de l'échantillonnage choisi et du paramètre δ utilisé lors de la création du graphe. La figure suivante (Figure 3.20) illustre ce phénomène. L'intersection peut être représentée par deux ou trois nœuds, suivant la distance d'acquisition de l'image. Dans ce cas, l'appariement point à point ne propose pas de solution satisfaisante. Il faudrait pouvoir assigner plusieurs nœuds à un seul, ce qui n'est pas possible avec notre approche par la méthode Hongroise car $P \in Per(|N_1|, |N_2|)$. Cependant, il est possible de détecter ce phénomène lors du matching par projection. Cela rend le matching plus cohérent mais la fusion des nœuds devient très complexe et n'a pas été développée dans cette étude.

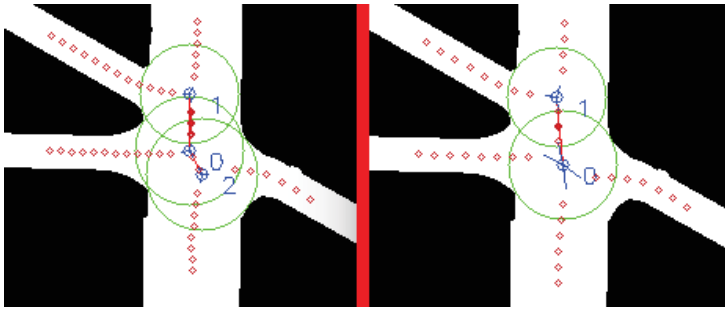


Figure 3.20. – Représentations topologiques différentes de la même intersection.

Notons enfin que le matching est d'autant meilleur que le taux de chevauchement entre les graphes est élevé. Ceci peut être le cas au début de la mission lorsque les deux robots ne sont pas très éloignés. Si le premier matching est correct, la fusion sera concluante et les matchings suivants auront plus de chance d'être réussis. Si le matching n'est pas concluant, la fusion n'est pas effectuée et le matching peut être retenté lorsque les cartes contiennent plus d'informations « communes ».

Dans la suite, on s'intéresse à la fusion de ces cartes hétérogènes.

3.5. Fusion des cartes

En supposant le matching résolu, nous nous intéressons maintenant à la fusion des deux cartes. Pour cela, la stratégie consiste à ne conserver, dans la carte résultante, que les informations les plus pertinentes de chacune des cartes.

Licence CC BY-NC-ND 3.0

3.5.1. Stratégie adaptée au contenu des cartes

La complémentarité des points de vue permet de créer une carte augmentée qui apporte de nouvelles informations au robot au sol mais également au drone. Dans cette étude, nous allons nous concentrer sur l'apport des informations provenant de la carte aérienne pour le robot terrestre. Ici, la carte fusionnée a pour but d'être utilisée par un robot terrestre pour la navigation et la planification de trajectoires.

Carte créée par un drone

Le point de vue aérien du drone lui permet d'avoir ou de construire une vision globale de l'environnement. La position des nœuds est donc fiable. Cependant, l'altitude du drone rend les informations locales moins précises que celles du robot terrestre.

Carte créée par un robot terrestre

Les informations locales sont plus précises lorsqu'elles sont acquises localement par le robot terrestre, elles doivent donc être privilégiées pour que la carte résultant de la fusion soit précise. De plus, pour permettre des fermetures de boucles fiables par le robot terrestre, il est préférable de conserver les perceptions terrestres. La dérive de l'odométrie peut déformer la carte, et une correction de la position des nœuds lors de la fusion est préférable.

Informations contenues dans la carte fusionnée

En conclusion, le repère conservé pour la carte globale est celui de la carte aérienne ainsi que la position des nœuds. Si un nœud est matché, ses informations géométriques sont celles du nœud de la carte terrestre. Les arêtes sont celles perçues par le drone, et complétées par celles perçues localement par le robot terrestre.

3.5.2. Carte augmentée construite par projection

La carte augmentée est créée en dupliquant la carte aérienne à l'échelle du robot terrestre puis en projetant les informations des nœuds matchés de la carte terrestre. La projection est la même que lors du matching par projection (sous-section 3.3.5) : la transformation rigide H_{n_1}est construite à partir des positions des nœuds n_1 et n_2, de l'angle de rotation $\alpha(n_1, n_2)$ et du facteur d'échelle $\lambda(n_1, n_2)$. Une sécurité est mise en place pour éviter l'ajout d'informations erronées. Les informations ne sont ajoutées que si l'angle de rotation $\alpha(n_1, n_2) \in [\overline{\alpha} \pm 3\sigma_\alpha]$ et le facteur d'échelle $\lambda(n_1, n_2) \in [\overline{\lambda} \pm 3\sigma_\lambda]$. Autrement, les informations locales sont considérées incohérentes et ne sont pas ajoutées. Les nœuds conservent une indication de la source de l'information (carte terrestre ou aérienne).

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

La figure 3.21 est un exemple de carte fusionnée. Les cercles orange représentent les informations provenant de la carte aérienne et les cercles marron proviennent de la carte terrestre. L'information locale du nœud central (indiqué par une flèche rouge) n'a pas été ajoutée car elle ne satisfaisait pas la condition de cohérence globale. On peut remarquer que le repère global est celui de la carte aérienne. La carte terrestre qui était déformée par les erreurs d'odométrie est rectifiée. Les informations provenant des deux sources ne se superposent pas parfaitement. C'est particulièrement le cas dans l'ellipse rouge. Il y a plusieurs raisons à cela : premièrement, comme pour la cohérence globale, les erreurs d'odométrie peuvent faire dévier les courbures des arêtes. Deuxièmement, l'hypothèse du monde plan n'est pas valable partout. À certains endroits, il y a une légère élévation qui crée une différence entre la distance parcourue et la distance mesurée dans l'image orthonormale. Dans ce cas, les informations terrestres sont plus pertinentes que les informations aériennes puisqu'elles correspondent à ce que le robot au sol perçoit réellement. Elles sont essentielles pour des fermetures de boucles robustes.

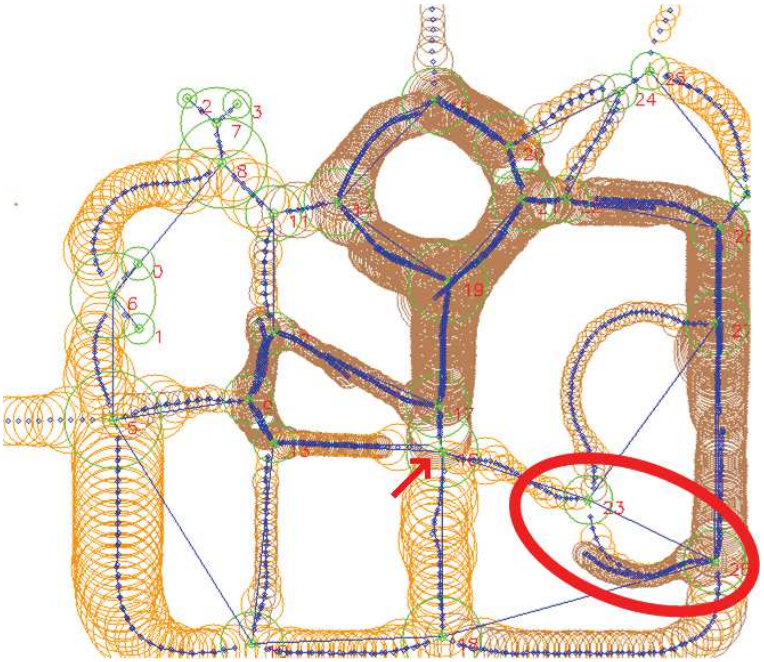


Figure 3.21. – Carte augmentée obtenue par fusion des cartes aérienne et terrestre. Les cercles orange représentent les informations provenant de la carte aérienne et les cercles marron proviennent de la carte terrestre. La flèche indique un nœud terrestre non projeté et la zone entourée correspond à une zone où les informations aériennes et terrestres diffèrent à cause d'une inclinaison du sol.

3.6. Utilisation de la carte fusionnée

Le matching et la fusion de données aéro-terrestres nous permettent de superposer les cartes et de faire une sélection pour ne conserver que les informations les plus pertinentes. Cette nouvelle carte ainsi que les informations obtenues lors du matching peuvent être utilisées de plusieurs manières qui ne sont pas exclusives.

Localisation et navigation terrestre

La nouvelle carte n'est pas définie dans le repère du robot terrestre. Il faut repositionner le robot terrestre dans cette nouvelle carte (localisation globale). Pour cela, on utilise la projection locale (associée au nœud matché le plus proche) appliquée à la pose du robot. Cette projection est peu précise, mais si le robot terrestre se trouve assez près d'une intersection matchée, elle peut être considérée comme valide. De plus, le robot peut toujours détecter une fermeture de boucle même en utilisant les informations provenant du drone. Cela lui permet de se relocaliser par rapport à ce nœud dans la carte fusionnée. L'illustration suivante (Figure 3.22) présente ce cas de figure où le robot est relocalisé grâce au nœud le plus proche (en violet). Cela permet la prédiction d'une fermeture de boucle avec un nœud qu'il n'a pas encore visité mais que le drone a cartographié.

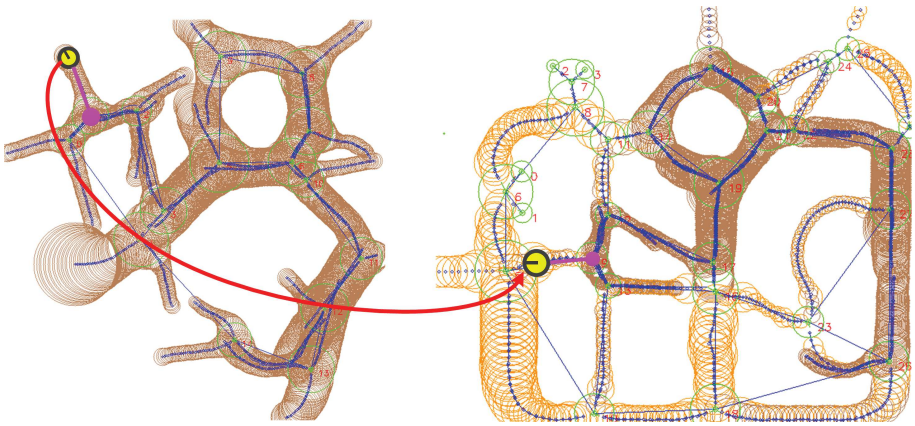


Figure 3.22. – Localisation et navigation terrestre à l'aide de la carte fusionnée. Carte terrestre (à gauche). Carte augmentée (à droite). Le robot terrestre est représenté par un cercle orienté en jaune. La relocalisation est calculée grâce au nœud proche en violet.

Cartographie collaborative intérieure et extérieure

Une autre utilisation est de profiter de la complémentarité des robots hétérogènes pour créer une carte comprenant à la fois les espaces navigables autour d'un bâtiment ainsi que ceux à l'intérieur du même bâtiment. Pour cela, les informations accumulées par le robot terrestre à l'intérieur sont ajoutées à la carte aérienne de l'extérieur

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

du bâtiment. Cela n'est possible que si le robot a également exploré une partie de l'extérieur du bâtiment. L'algorithme a en effet besoin d'un chevauchement des cartes pour fonctionner.

Extension à plus de deux robot

Avec l'approche proposée, il n'est pas possible de matcher plus de deux cartes en même temps. Il est néanmoins possible d'obtenir des fusions successives entre les robots pour obtenir une carte globale et de la propager au sein de tous les robots.

Cela permet également une localisation de tous les acteurs d'une mission dans un référentiel commun.

Coordination de mission aéro-terrestre

Grâce à la fusion des cartes, des éléments importants pour une mission sont propagés d'une carte à l'autre. Ces éléments peuvent être, entre autres, des cibles à atteindre pour le robot terrestre, où des lieux d'atterrissage pour le drone. La position de ces éléments et celles des robots se retrouvent exprimées dans un même repère facilitant ainsi la coordination de la mission aéro-terrestre.

GPS émulé

Si un seul des robots est équipé d'un système de localisation GPS, il peut ajouter cette information de position GPS aux nœuds qu'il a détectés. Ces informations sont conservées dans la carte fusionnée et peuvent être utilisées par l'autre robot qui ne dispose pas de localisation absolue pour simuler ce dispositif.

Ce type de coopération aéro-terrestre est testé dans [Frietsch et al., 2008]. Un drone équipé d'un GPS propage la localisation GPS à un robot au sol grâce à une estimation de sa position relative en utilisant une caméra. Une expérience similaire utilisant notre approche avec une carte aérienne et une carte provenant de OpenStreetMap sera présentée dans le chapitre suivant.

Conclusion

Ce chapitre présentait la contribution principale de la thèse : la coopération aéro-terrestre pour la localisation et la cartographie. Après avoir proposé un modèle unifié pour la représentation de l'environnement à partir de différents points de vue aérien et terrestre, nous avons développé une approche hiérarchique pour la mise en correspondance et la fusion de ces cartes. Cette association de nœuds est effectuée grâce à une mesure de dissimilarité entre les descripteurs de ces nœuds. La mesure est construite en essayant de respecter les critères de discriminance et d'invariance aux éléments inconnus. Un score est mis en place pour déterminer la qualité et la vraisemblance du matching. S'il est correct, le matching est utilisé pour fusionner les

Licence CC BY-NC-ND 3.0

deux cartes en une unique carte globale augmentée. Cette carte fusionnée contient les informations les plus pertinentes des deux sources. Le robot terrestre est relocalisé dans cette carte pour qu'il puisse détecter des fermetures de boucle en exploitant les informations aériennes ajoutées et continuer sa mission. La coopération ne se limite pas uniquement à la fusion de cartes, elle peut également être utilisée pour propager des informations GPS ou des lieux d'intérêt.

Cette approche a été testée lors de plusieurs réalisations expérimentales, que nous présenterons dans le chapitre suivant.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

4. Évaluations expérimentales

Sommaire

Introduction **92**
4.1. Présentation des sites d'expérimentations **92**
4.1.1. Plateforme expérimentale PAVIN 92
4.1.2. Parking à proximité du laboratoire 94
4.1.3. Hall d'essais de l'ENSIL-ENSCI 95
4.1.4. Étage de l'ENSIL-ENSCI 95
4.1.5. Expérimentations à l'ETH de Zurich 96
4.1.6. Environnements de développement 97
4.2. Création de cartes aériennes **98**
4.2.1. Détection de l'espace navigable 98
4.2.2. Gestion de la visibilité partielle de l'environnement . . . 102
4.2.3. Gestion des éléments dynamiques 105
4.3. Évaluation de la cartographie coopérative **108**
4.3.1. Impact des paramètres de la mesure 108
4.3.2. Évaluation de la mise en correspondance des cartes . . . 116
4.3.3. Localisation à partir de la carte fusionnée 120
4.4. Utilisation de cartes en ligne **122**
4.4.1. Création de cartes géoréférencées 123
4.4.2. Matching des cartes OSM et aérienne 124
Discussion **127**

Introduction

Ce chapitre regroupe des expérimentations réalisées pour évaluer les différentes parties de la cartographie coopérative aéro-terrestre proposée. La création d'une carte de l'espace navigable terrestre à l'aide d'un drone peut être difficile lorsque l'environnement est peu structuré. Des obstructions visuelles peuvent également compromettre l'intégrité de la carte, de même que la présence d'éléments dynamiques. La coopération aéro-terrestre se fait en deux étapes qu'il est important d'évaluer : la fusion des cartes et l'utilisation par le robot terrestre de la carte ainsi créée.

Dans un premier temps, nous présentons les différents sites d'expérimentations, leurs dimensions et leurs caractéristiques. Puis, nous introduisons rapidement l'environnement logiciel utilisé pour le développement des algorithmes, en particulier l'utilisation du middleware ROS et du simulateur 4DVirtualiz. Enfin, les expérimentations sont présentées avec leurs objectifs et résultats.

4.1. Présentation des sites d'expérimentations

Les environnements considérés sont de différents types, ils peuvent être intérieur ou extérieur, réel ou simulé. Les robots utilisés dans ces environnements n'ont pas toujours les mêmes caractéristiques. Ils permettent d'évaluer les capacités de l'approche proposée et son adaptabilité au contexte.

4.1.1. Plateforme expérimentale PAVIN

La plateforme PAVIN [1] est un site expérimental universitaire situé à Clermont-Ferrand (Figure 4.1). Ce site est un mini environnement urbain de 5000m^2 avec des bâtiments et des décors. Il est équipé de feux tricolores et de signalisations au sol ou à l'aide de panneaux. Nous disposons d'un modèle 3D réaliste de cet environnement. Ce site est l'environnement simulé que nous avons utilisé (Figure 4.2).

1. Plateforme Auvergne pour Véhicules INtelligents

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Figure 4.1. – Site expérimental PAVIN.

Le site est constitué d'un réseau routier avec des intersections et un rond-point. Les décors sont suffisamment hauts pour bloquer la vue des robots au sol mais pas trop hauts pour obstruer la vision de la route depuis le ciel. Ce site a l'avantage de proposer peu d'intersections similaires mais le sol présente des élévation à certains endroits, ce qui met parfois en défaut l'hypothèse du monde plan.

A l'aide du simulateur, nous pouvons intégrer des drones et des robots terrestres équipés de capteurs. Le simulateur met à disposition entre autres un modèle de quadcopter et le modèle du robot terrestre *Pioneer 3-AT*. Les capteurs embarqués sont une caméra perspective et une caméra omnidirectionnelle catadioptrique avec un miroir hyperbolique. Cet environnement a servi pour le développement des algorithmes de cartographie aérienne, terrestre et pour la cartographie coopérative. Il a également permis de tester la fusion de cartes provenant d'une base de données car ses routes sont disponibles dans OpenStreetMap.

Figure 4.2. – Modèle 3D du site expérimental PAVIN.

Licence CC BY-NC-ND 3.0

4.1.2. Parking à proximité du laboratoire

Pour avoir un environnement urbain réel pour la construction de cartes aériennes, plusieurs vols ont été effectués au dessus d'un espace présentant un réseau routier particulièrement structuré. La zone cartographiée fait environ 50m sur 40m.

Figure 4.3. – Vue de drone du parking.

Ce parking construit récemment est un environnement adéquat pour la détection de routes. Le contraste entre la route et les trottoirs ou l'herbe facilite la classification de l'espace navigable. Les véhicules roulants sur ce parking ont permis de tester la création d'une carte en présence d'éléments dynamiques.

Cet environnement n'a été exploré qu'avec un drone, le *Bebop2* développé par *Parrot*. Ce drone est équipé de deux caméras. Une caméra orientée vers le sol qui sert à l'asservissement en position et dont le flux vidéo n'est pas accessible. Une deuxième caméra orientée vers l'avant qui peut être inclinée légèrement vers le sol. Cette caméra a l'avantage d'être stabilisée numériquement, ainsi les mouvements du drone ne font pas varier l'angle de vue.

Figure 4.4. – Drone *Bebop 2* (*Parrot*), utilisé dans les expérimentations.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

4.1.3. Hall d'essais de l'ENSIL-ENSCI

Pour tester la coopération aéro-terrestre au delà de la simulation, un espace navigable restreint a été construit dans un hall d'essais de l'ENSIL (école nationale supérieure d'ingénieurs de Limoges).

Figure 4.5. – Environnement de test de la coopération aéro-terrestre.

Cet environnement a pour but de tester la cartographie coopérative hétérogène, c'est-à-dire, la fusion de cartes et la navigation du robot terrestre. Pour se concentrer sur ce problème, la segmentation de l'espace navigable y est aisée. Les frontières blanches se distinguent facilement du sol gris. L'espace modulaire permet de tester différentes configurations géométriques.

Les robots utilisés, visibles sur la photo (Figure 4.5), sont le *Bebop2* avec sa caméra perspective et le *Pioneer 3-AT* avec une caméra omnidirectionnelle catadioptrique.

4.1.4. Étage de l'ENSIL-ENSCI

Les plans d'architecte du bâtiment ENSIL ont été utilisés pour construire la carte hybride et tester la classification sémantique. Ce type de plan nécessite un certain nombre de traitements pour générer la carte hybride.

Licence CC BY-NC-ND 3.0

Figure 4.6. – Plan de l'étage de l'ENSIL-ENSCI.

Cet environnement n'a pas été utilisé avec des robots réels. Il a permis cependant de mettre en évidence l'utilisation des données sémantiques et des représentations multiples pour définir des routes possiblement fermées (portes closes).

4.1.5. Expérimentations à l'ETH de Zurich

Ces expérimentations ont été faites avec l'aide du laboratoire *Autonomous System Lab* (ASL) de l'École Polytechnique Fédérale de Zurich (ETH). La première expérimentation a été réalisée dans les espaces verts près de l'ETH. La deuxième s'est déroulée aux alentours d'un bâtiment effondré à Genève.

Figure 4.7. – Espace vert à Zurich et bâtiment à Genève.

La coopération aéro-terrestre n'était pas le but de ces expérimentations (pas encore développée pendant la période des tests). Nous avons voulu tester l'utilisation des cartes topologiques dans des environnements très peu structurés. Le premier environnement est constitué d'arbustes et de bosquets qui rendent difficile la définition précise de l'espace navigable. Le second présente un monticule infranchissable entouré d'un grand espace vide. Dans ce cas, c'est la structure topologique qui fait défaut.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Le drone utilisé est un *Firefly* conçu par *Ascending Technologies*. Il est équipé de deux caméras stéréo *VIsensor* développées par l'équipe ASL.

4.1.6. Environnements de développement

Les expérimentations ont été réalisées sur un ordinateur portable avec un processeur Intel Core i7 @ 2.5GHz, 16GB RAM et le système d'exploitation Ubuntu14.4 (64-bit).

Robot Operating System

Les algorithmes présentés dans ce manuscrit ont été développés en C++ au sein de l'écosystème ROS. C'est un environnement qui propose des outils de visualisation et de paramétrage. Il facilite la mise en place de communication entre l'ordinateur et les robots. Cet environnement permet notamment d'enregistrer les expérimentations pour les rejouer et les analyser. ROS rassemble une grande communauté active qui met à disposition des ressources pour aider au développement de la robotique. La version installée est ROS Indigo. Pour plus de détails, nous renvoyons le lecteur vers le site *ros.org*.

Simulateur 4DVirutaliz

Le logiciel 4DVirtualiz est un simulateur développé par une jeune entreprise de Clermont-Ferrand. C'est un simulateur qui modélise les phénomènes physiques de manière réaliste (variation de l'éclairage, masquage des signaux GPS, etc). Les modèles dynamiques des objets sont également intégrés au simulateur. Ce simulateur est lancé sur un ordinateur dédié. L'intérêt de ce simulateur est que l'on peut connecter se connecter indifféremment au simulateur ou au système réel. Une interface TCP/IP permet d'envoyer les informations du simulateur à l'ordinateur. En particulier, les caméras fonctionnent comme des caméras IP, le flux est envoyé sur le réseau. Il est aussi possible de coupler dans le même scénario des éléments virtuel du simulateur et des objets physiques dans le monde réel. Les éléments de la simulation sont mis en place et contrôlés grâce à un système de programmation graphique facile à prendre en main (Figure 4.8).

Licence CC BY-NC-ND 3.0

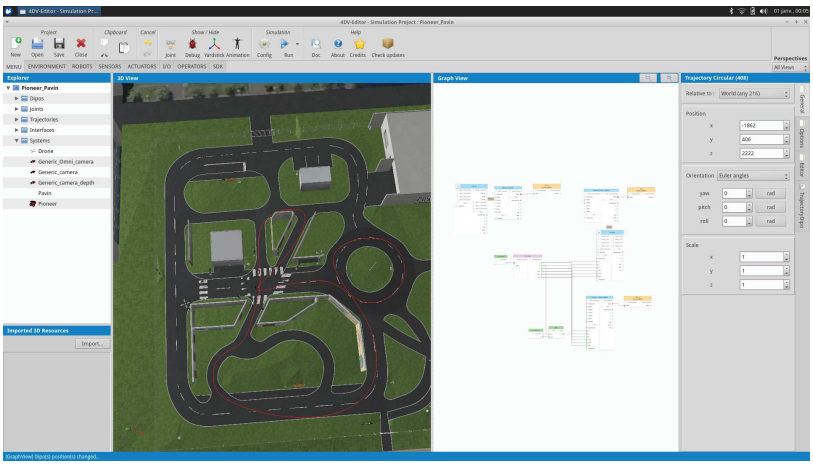


Figure 4.8. – Interface graphique du simulateur 4DVirtualiz.

4.2. Création de cartes aériennes

La création de la carte de traversabilité terrestre à partir de données acquises par le drone n'est pas évidente. Plusieurs phénomènes peuvent gêner la création de la carte. Tout d'abord, la détection de l'espace navigable peut être une difficulté. Ensuite, la gestion des obstructions ou de la vision partielle de l'environnement nécessite un traitement particulier. Enfin, la gestion des éléments dynamiques présents dans l'environnement impacte également la stabilité et la persistance de la carte. Nous avons testé la robustesse de notre cartographie aérienne face à ces phénomènes qui pourraient perturber la création de la carte.

4.2.1. Détection de l'espace navigable

La méthode de détection de l'espace navigable dépend des capteurs embarqués sur le drone. Les données d'observation peuvent être des images image 2D, stéréo ou un nuage de points provenant d'une reconstruction 3D.

4.2.1.1. Détection à partir d'une image aérienne

Pour que la détection de l'espace navigable soit géométriquement correcte, la caméra est préalablement calibrée et l'image est projetée dans le plan du sol.

Dans les expérimentations utilisant le simulateur (Pavin), la caméra est stabilisée en orientation vers le sol et les paramètres de calibration du capteur simulé sont

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

connus. Pour les tests au dessus du parking et dans le hall d'essais, le drone Bebop2 est équipé d'une caméra à l'avant orientée légèrement vers le sol. Comme expliqué précédemment, cette caméra est stabilisée numériquement, ainsi les mouvements du drone ne font pas varier l'angle de vue.

À l'aide d'une étape de calibration sur une zone prédéfinie, on peut calculer la transformation permettant de passer de l'image perspective à l'orthoimage. Les deux images (Figure 4.9) montrent une image prise par le drone et son orthoimage associée.

Figure 4.9. – Transformation en orthoimage. (gauche) Image acquise par le drone, (droite) Orthoimage correspondante.

Lorsque l'espace navigable est suffisamment contrasté par rapport aux autres éléments de la scène (typiquement, les environnements routiers), une segmentation couleur peut suffire à extraire l'espace navigable. C'est le cas dans les expérimentations sur le parking et le site Pavin, où les voies et passages piétons sont facilement séparables (Figure 4.10). Un lissage morphologique permet de réduire le bruit. La carte de traversabilité qui en résulte ne prend pas en compte le sens de la route ni les signalisations.

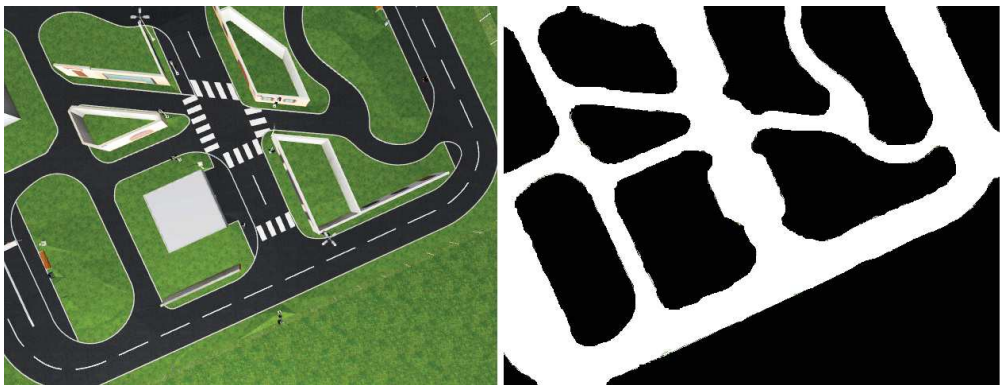


Figure 4.10. – Image aérienne provenant du simulateur et résultat de segmentation couleur de l'espace navigable.

Licence CC BY-NC-ND 3.0

Une autre solution est d'initialiser un contour fermé dans l'espace navigable et de « gonfler » cette forme jusqu'à atteindre les frontières d'obstacles et ainsi délimiter l'espace navigable. On peut notamment utiliser des algorithmes utilisant un front d'onde ou un contour actif [Merveilleux et al., 2011]. Cette méthode est également utilisée pour la détection de l'espace navigable dans l'image omnidirectionnelle du robot terrestre et pour la création de la carte en utilisant le plan d'un bâtiment. Elle permet de garantir la connexité de la carte.

4.2.1.2. Détection à partir d'une perception 3D de l'environnement

Une caméra stéréoscopique permet de reconstruire les informations 3D de l'environnement. Ces informations peuvent être utilisées pour calculer l'espace navigableen fonction des capacités de franchissement du robot. Comme présenté dans [Guastella et al., 2017], la capacité de traversabilité d'un robot peut être définie par un seuil de pente maximale. On estime alors la normale au sol pour déterminer l'angle d'inclinaison (Figure 4.11). Cela permet ensuite de segmenter l'espace selon cet angle pour créer la carte de traversabilité.

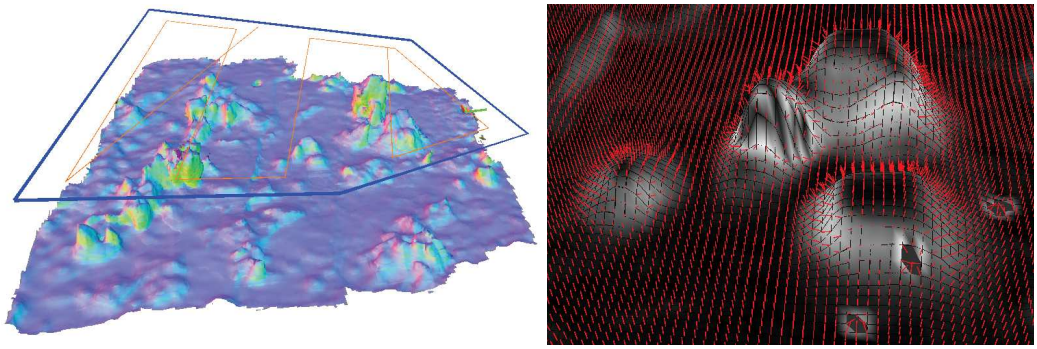


Figure 4.11. – (gauche) Carte 3D d'un environnement avec la trajectoire du drone, (gauche) normales au sol permettant de déterminer la traversabilité pour le robot.

Lorsque la cartographie est un nuage de points ou un maillage 3D, la bibliothèque *(Point Cloud Library)* permet de calculer les normales à un point en utilisant son voisinage. Les images suivantes (Figure 4.12) présentent la cartographie 3D du bâtiment effondré à Genève ainsi que l'espace navigable estimé grâce aux normales au sol (en bleu foncé dans l'image de droite).

Lorsque la représentation est une grille d'élévation, la normale au sol peut être estimée à l'aide d'un calcul de gradient dans l'image correspondant à l'altitude. La figure 4.13 présente la cartographie de l'espace vert proche de l'ETH. Le dégradé du violet vers le rouge montre l'altitude et la nappe en transparence bleutée est la carte de traversabilité superposée.

Les cartes de traversabilité d'environnements 3D ont été utilisées pour tester la validité d'une carte topologique dans un environnement peu structuré. Les cartes 3D

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

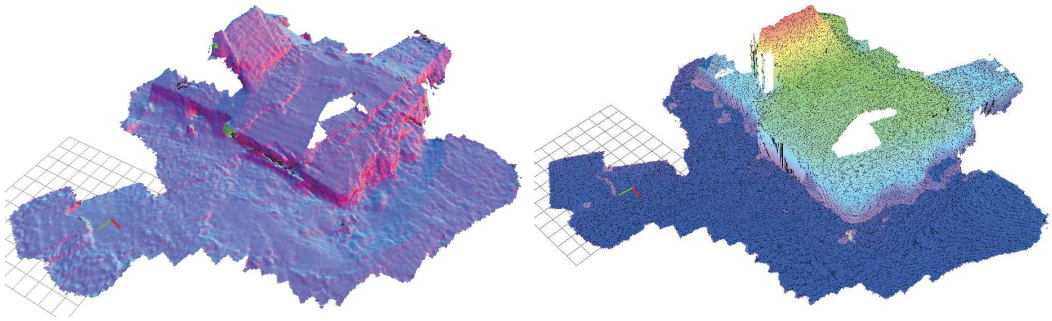


Figure 4.12. – Carte 3D du bâtiment (à gauche) et l'espace navigable estimé (à droite).

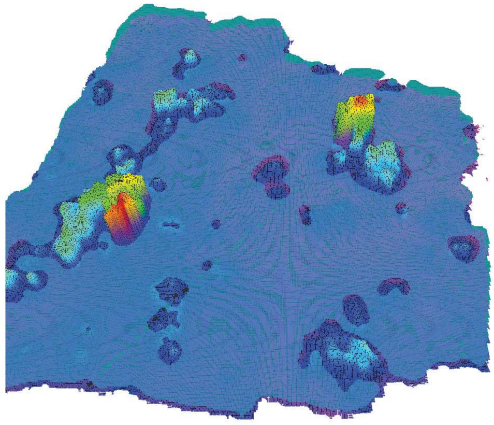


Figure 4.13. – Grille d'élévation avec l'espace navigable en transparence.

sont construites à l'aide de l'outil *Voxblox* [Oleynikova et al., 2017] développé par l'équipe ASL. Les environnements peu structurés où l'espace navigable est difficile à définir présentent un problème de variabilité et d'instabilité de la représentation. En effet, un léger bruit de mesure peut faire changer la forme d'un obstacle ou en faire apparaitre, ce qui perturbe beaucoup la représentation topologique. Les images suivantes (Figure 4.14) montrent trois cartes de traversabilité avec leur carte hybride pour le même espace vert. Dans ce cas, les cartes hybrides ne peuvent pas être matchées, car les représentations topologiques sont trop différentes. Il y a essentiellement deux raisons à ce problème. Premièrement, la squelettisation n'est pas une fonction stable, elle n'est que semi-continue. L'apparition ou la disparition d'un obstacle même très petit peut changer complètement la forme du squelette. Deuxièmement, le calcul de la traversabilité dans cet environnement "herbeux" n'est pas stable non plus. Plusieurs survols vont engendrer plusieurs représentations différentes, le bruit de mesure et les paramètres de calcul des normales vont également modifier la carte de traversabilité. Il est important que la détection de l'espace navigable soit la plus stable possible pour garantir des cartes similaires.

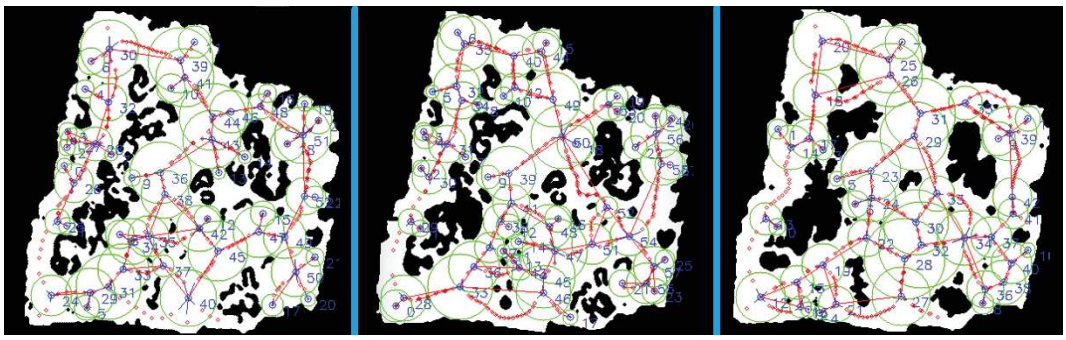


Figure 4.14. – Variation de la carte hybride d'un même lieu.

Ces méthodes nécessitent d'avoir la carte de l'environnement puis de construire la carte hybride à partir de cette représentation 2.5 ou 3D. Dans ce type d'environnements, la mise en correspondance n'est pas évidente, voire impossible dans certaines situations. Cependant, la cartographie aérienne peut servir à déterminer l'accessibilité d'un lieu pour le robot terrestre.

4.2.2. Gestion de la visibilité partielle de l'environnement

Dans des environnements à grande échelle, la perception d'un drone est limitée et on fait habituellement appel à des algorithmes de SLAM pour cartographier l'intégralité de la zone à explorer. La perception, en plus d'être partielle, peut souffrir d'obstructions à cause des obstacles tels que les bâtiments ou les arbres qui empêchent de percevoir clairement l'espace navigable. Dans le cas des représentations 3D comme vu précédemment, cela n'est pas un problème la cartographie s'appuie sur la position 3D des obstacles et des frontières. Par contre, dans le cas de l'utilisation d'images, les informations sont obtenues par projection en utilisant l'hypothèse d'un monde plan, ce qui peut créer des erreurs de cartographie. Pour la création des cartes aériennes à partir d'images partielles, plusieurs solutions ont été testées.

4.2.2.1. Création d'une mosaïque d'images

Dans un premier temps, la création incrémentale d'une mosaïque d'images aériennes a été envisagée, mais le résultat n'était pas satisfaisant. La construction incrémentale d'une mosaïque d'images ne fonctionne que si l'hypothèse d'un monde plan est valide ou si les images sont rectifiées. Dans le cadre de nos expérimentations, la vision par drone à une altitude inférieure à 50m rend cette hypothèse erronée pour les bâtiments. Les obstacles sont projetés au mauvais endroit et la carte est faussée.

La solution est de construire la mosaïque de manière globale et non incrémentale. On peut ainsi n'utiliser que les images qui génèrent le moins d'occultations. Pour appliquer cette solution, nous avons utilisé l'algorithme *map2DFusion* proposé dans

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

[Bu et al., 2016]. En exploitant toutes les images prises par le drone, l'algorithme détecte les recouvrements en utilisant des points d'intérêt puis fusionne les informations dans les zones de chevauchement. Les occultations sont ainsi réduites si les prises de vue le permettent. Le résultat de cet algorithme est présenté dans l'image suivante (Figure 4.15).

On peut alors appliquer le processus de détection de l'espace navigable sur cette image globale et construire la carte hybride. De même que pour les approches 3D précédentes, la carte hybride est construite à partir d'une carte complète. Elle nécessite d'avoir toutes les images pour créer la carte de traversabilité globale.

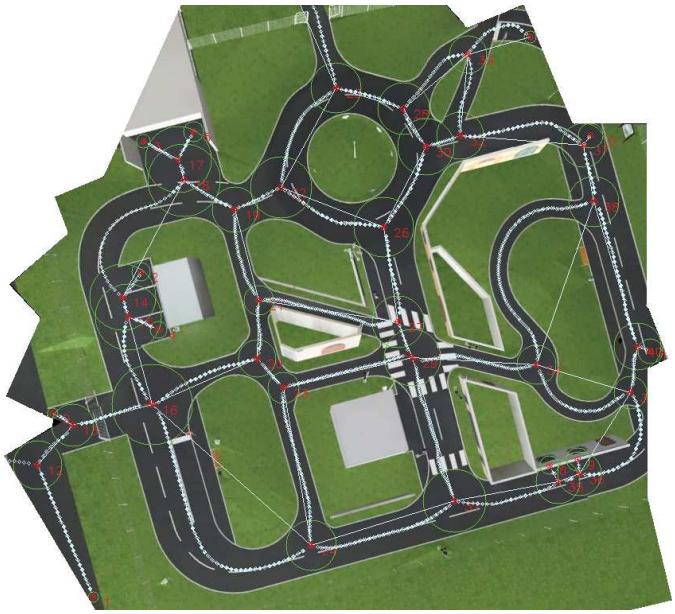


Figure 4.15. – Carte hybride construite sur une image globale du site experimental Pavin, obtenue par une mosaïque d'images.

4.2.2.2. Création d'une mosaïque de graphes

La seconde solution proposée est de construire une mosaïque de graphes et non d'images [Renaudeau et al., 2018]. De la même manière que pour la coopération aéro-terrestre, deux cartes hybrides sont matchées puis fusionnées. Il s'agit là de cartes hybrides aériennes construites à partir d'images successives du même drone. Les problèmes liés à la différence de perception sont donc grandement réduits. Une mesure de dissimilarité adaptée est calculée entre les nœuds des deux graphes puis ils sont matchés en utilisant l'algorithme Hongrois.

$$D(n_1, n_2) = \begin{array}{l} K_d \left| d_{n1} - d_{n2} \right| + K_r \left| R_{n1} - R_{n2} \right| + \\ \min\limits_{\alpha} \sum\limits_{a1 \in [1, d_{n1}]} \left[\min\limits_{a2 \in [1, d_{n2}]} K_a (\theta_{a1} - (\theta_{a2} + \alpha)) + K_l \left| l_{a1} - l_{a2} \right| \right] \end{array} \quad (4.1)$$

Licence CC BY-NC-ND 3.0

Cette mesure de dissimilarité est semblable à celle utilisée pour le matching aéro-terrestre (Équation 3.3). On peut y voir les deux termes correspondant à la différence des degrés des nœuds et la différence des angles de branches. Deux autres termes ont été ajoutés. La différence entre les rayons des nœuds devient un élément discriminant utilisable car les échelles des cartes sont similaires (identique dans le cas de vol à altitude fixe). On ajoute également la différence entre les longueurs des branches l_{a1} et l_{a2} pour la même raison. La différence n'est plus calculée pour $\alpha \in [0, 2\pi]$ mais seulement pour les angles d'incidence des arêtes : $\alpha \in \{\theta_{a2} \mid a2 \in [1, d_{n2}]\}$. Cette formulation est beaucoup plus rapide à calculer car on ne compare pas la « forme » complète des nœuds mais seulement les angles d'incidences et la longueur des arêtes pour quelques orientations.

Le raffinement, la détection des fausses assignations et la validation du matching sont obtenus en calculant la transformation rigide entre les deux graphes à l'aide d'un algorithme RANSAC. Cette méthode permet de détecter les nœuds mal assignés car ils ne satisfont pas le critère de projection. Le processus d'assignation par consensus de projection est remplacé par une assignation au nœud le plus proche après projection. Le taux de chevauchement élevé entre deux images successives facilite le matching et diminue énormément le nombre d'assignations erronées. Cette méthode est beaucoup plus rapide que celle utilisée pour la fusion aéro-terrestre et peut être effectuée en ligne. Pour construire cette mosaïque de graphe, le drone n'est pas obligé de voler à une altitude fixe. Un facteur d'échelle variable est constamment recalculé après chaque matching pour conserver la cohérence de la carte globale quelle que soit l'altitude du drone.

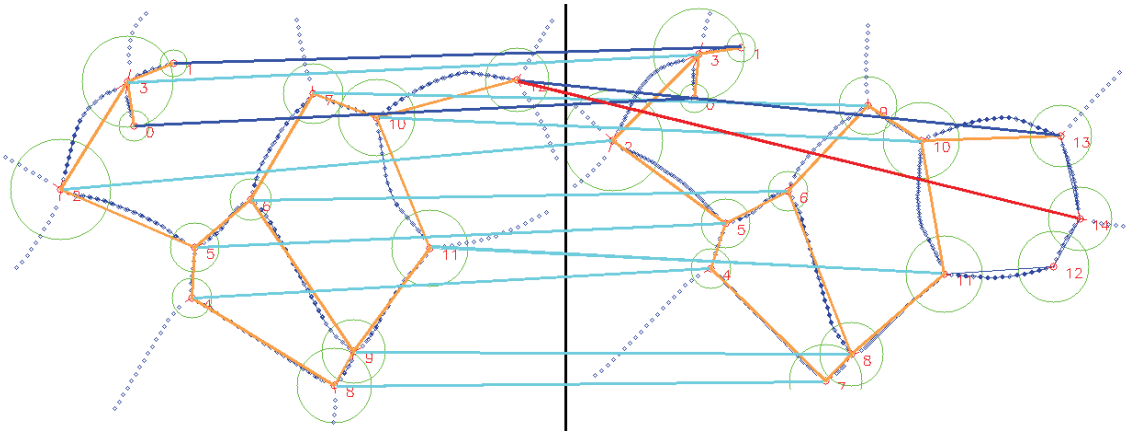


Figure 4.16. – Matching de cartes aériennes locales. Les traits bleu clair entre les deux graphes montrent les bonnes assignations et les traits rouges montrent les assignations erronées.

La figure 4.16 représente un résultat de matching entre deux cartes successives. Les traits bleu clair montrent le matching initial, le trait rouge indique la fausse association détectée par l'algorithme RANSAC, et les traits bleu foncé correspondent aux assignations ajoutées grâce à la projection. Les structures topologiques en orange

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

sont détectées comme partie commune entre les deux graphes. Comme dans la fusion aéro-terrestre, les nouveaux nœuds sont ajoutés à la carte globale. Ainsi, la carte globale est construite de manière incrémentale. La carte globale créée par cette méthode est similaire à celle créée par mosaïque d'image mais elle ne nécessite pas l'enregistrement des images tout au long de l'exploration. Dans les deux méthodes, les obstacles peuvent perturber la cartographie. Une gestion des éléments dynamiques est donc mise en place pour corriger ce problème.

4.2.3. Gestion des éléments dynamiques

Des éléments dynamiques tels que des piétons ou des véhicules peuvent perturber la création de la carte. Prenons l'exemple d'un véhicule se déplaçant sur une route, la carte créée est localement modifiée. Des nœuds sont ajoutés autour de l'élément dynamique. Il faudrait pouvoir détecter que ces nœuds créés sont des artéfacts.

Pour cela, nous associons une donnée de confiance aux nœuds lors de leur création. Lors de matchings successifs des cartes aériennes, la carte globale est créée et l'information de confiance des nœuds est mise à jour. Si un nœud est matché à plusieurs reprises, son indice de confiance augmente. Inversement, si ce nœud n'a été vu que très rarement, son indice de confiance diminue, jusqu'à ce que le nœud soit supprimé lorsque l'indice de confiance devient trop faible.

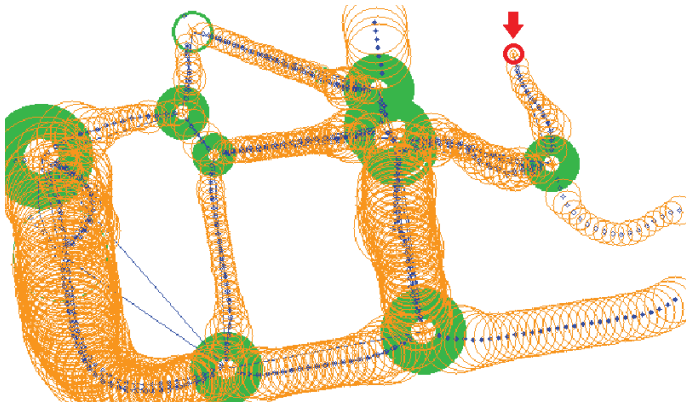


Figure 4.17. – Indice de confiance des nœuds. La complétion des disques représentant les nœuds indique leur niveau de confiance. Le nœud en rouge possède une valeur de confiance trop faible et sera supprimé.

Le déplacement d'un obstacle génère ainsi des nœuds à plusieurs emplacements différents. Ces nœuds ne seront plus détectés dans les cartes suivantes, leur indice de confiance va donc diminuer jusqu'à leur suppression. La carte Figure 4.17 montre l'indice de confiance sous forme de disque plus ou moins rempli selon la valeur de

Licence CC BY-NC-ND 3.0

cet indice. Le nœud pointé en rouge a un indice de confiance en dessous du seuil critique et sera supprimé.

Cette méthode a été testée pour les observations partielles sur Pavin et dans l'environnement réel du parking pour les éléments dynamiques. Des véhicules ont circulé sur la route mais n'ont perturbé la carte globale que pendant un court moment (Figure 4.18). L'image centrale (a) montre l'image capturée par le drone. La carte de traversabilité obtenue pour cette image est à gauche (b), on peut y voir que les voitures perturbent la détection de l'espace navigable puisque les chemins ne se prolongent pas. La carte de traversabilité obtenue après le traitement des nœuds fictifs créés par les voitures est présentée à droite (c). Les nœuds fictifs ont été supprimés et les chemins sont considérés comme traversables.

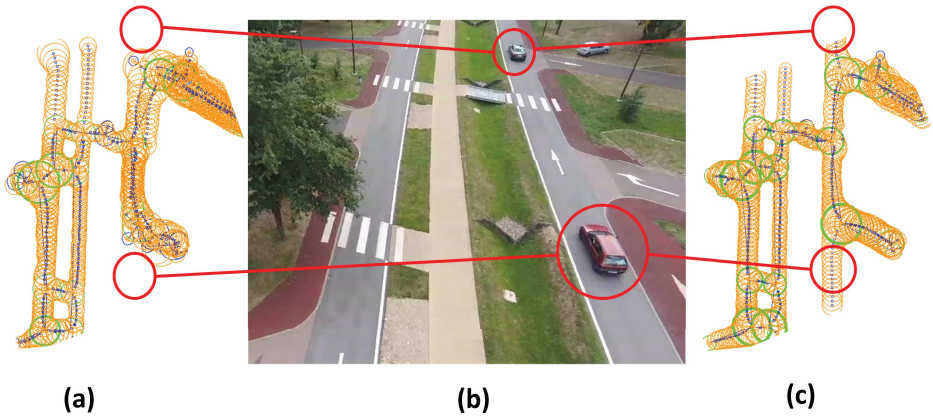


Figure 4.18. – Véhicules mobiles lors de la cartographie. (a) Carte de traversabilité perturbée par les véhicules, (b) Image capturée par le drone ,(c) Carte de traversabilité après la suppression des nœuds fictifs.

Cette gestion des éléments dynamiques n'a pas été mise en place pour la cartographie terrestre car les nœuds sont ajoutés lorsque le robot y passe (intersections). La carte est donc moins sensible aux éléments dynamiques.

Notons que contrairement à la mosaïque d'images, la mosaïque de graphe permet plusieurs représentations topologiques du même objet physique en même temps et au même endroit. Cette propriété a plusieurs effets. Premièrement, l'indice de confiance permet d'éliminer les représentations qui ne sont pas stables dans le temps parmi les multiples représentations. Deuxièmement, lors du matching aéro-terrestre, cela permet d'avoir plusieurs hypothèses de représentations possibles pour l'assignation des nœuds. Cette particularité est détaillée dans la section suivante.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

L'algorithme complet de la création de la carte aérienne est présenté dans le logigramme suivant (Figure 4.19) :

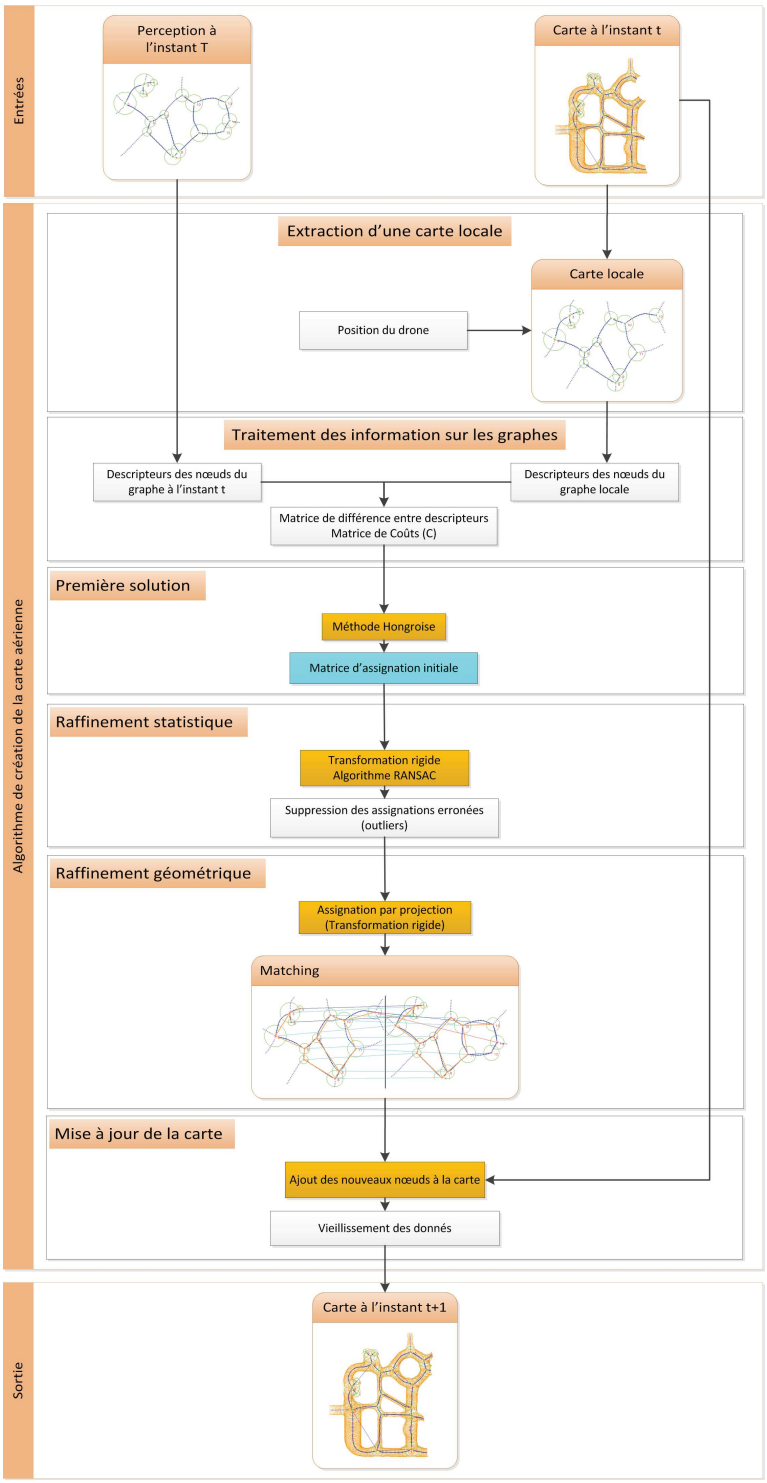


Figure 4.19. – Logigramme de l'algorithme de matching de graphes.

Licence CC BY-NC-ND 3.0

4.3. Évaluation de la cartographie coopérative

Nous avons évalué notre approche de la cartographie coopérative aéro-terrestre sur plusieurs critères. Pour l'étape de matching, nous nous sommes intéressés à la sensibilité de la mesure par rapport au réglage des coefficients de pondération, ainsi qu'au nombre minimal de nœuds en communs pour que le matching soit correct. L'effet des représentations multiples présentes dans la carte aérienne est également évalué.

Nous nous sommes concentrés sur l'utilisation de la carte augmentée par le robot terrestre. Nous avons testé les capacités du robot terrestre à se localiser dans cette carte lors de la reconnaissance de lieux (fermeture de boucle), en utilisant les nœuds provenant de sa propre carte ou de la carte aérienne.

Ces tests ont été effectués sur l'environnement simulé du site Pavin et en conditions réelles dans le hall d'essais de l'ENSIL-ENSCI.

4.3.1. Impact des paramètres de la mesure

L'impact des coefficients de pondération dans la mesure de dissimilarité a été évalué en faisant varier la valeur d'un seul paramètre à la fois et en regardant le nombre de fausses assignations produites par ce changement et la capacité du matching hiérarchique à rester correct. Cette évaluation a été faite sur plusieurs couples de cartes aéro-terrestre pour étudier l'évolution des cartes de couts lors de la variation des pondérations. Pour illustrer nos propos, nous utiliserons le matching aéro-terrestre qui a servi d'exemple tout au long du chapitre précédent.

Rappelons les mesures de dissimilarité initiale et raffinée du matching, respectivement :

$$D(n_1,n_2) = \begin{array}{l} K_d\,|d_{n1}-d_{n2}| + \\ \min\limits_{\alpha\in[0,2\Pi]} \left[K_c\,Eq_\alpha(n1,n2) + K_a \sum\limits_{a1\in[1,d_{n1}]} \min\limits_{a2\in[1,d_{n2}]} (\theta_{a1}-\theta_{a2}+\alpha) \right] \end{array} \tag{4.2}$$

$$D_2(n_1,n_2) = \begin{array}{l} K_d\,|d_{n1}-d_{n2}| + K_\lambda\left|\overline{\lambda}-\lambda\right| + K_\alpha\,|\overline{\alpha}-\alpha| + \\ \min\limits_{\alpha\in[\overline{\alpha}\pm 3\sigma_\alpha]} \left[K_c Eq_\alpha(n1,n2) + K_a \sum\limits_{a1\in[1,d_{n1}]} \min\limits_{a2\in[1,d_{n2}]} (\theta_{a1}-\theta_{a2}+\alpha) \right] \end{array} \tag{4.3}$$

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Les valeurs des pondérations utilisées par défaut sont présentées dans le tableau suivant (Tableau 4.1) :

$\Delta\alpha$	K_d	K_c	K_a	K_λ	K_α	C_{max}
2°	20	$\frac{1}{20}$	$\frac{1}{5}$	20	$\frac{40}{\sigma_\alpha}$	100

Table 4.1. – Liste des paramètres utilisés.

Variation du pas d'échantillonnage $\Delta\alpha$ de la mesure de dissimilarité

Ce paramètre permet de changer la résolution de calculs effectués pour déterminer la mesure de dissimilarité. Il est directement relié au temps de calcul, puisque le calcul de la matrice de couts occupe la plus grande partie du temps de calcul. Ce paramètre détermine également la précision des première et deuxième mesures de dissimilarité, car les valeurs d'angle minimales seront utilisées pour raffiner la mesure.

La figure 4.20 présente l'évolution de la carte de couts avec le paramètre $\Delta\alpha$. Les assignations en vert sont les assignations correctes et celles en rouge sont les erreurs à la sortie de l'algorithme Hongrois. Les assignations en orange représentent les assignations manquantes à la sortie de l'algorithme complet. S'il n'y a pas d'assignation en orange cela signifie que que les assignations initiales erronées (en rouge) ont été corrigées par les différentes étapes de l'algorithme. On constate que le paramètre $\Delta\alpha$ peut être réglé assez élevé ($\Delta\alpha$=21°) sans modifier l'exactitude du résultat final. Le nombre d'assignations incorrectes augmente avec le pas d'échantillonnage. Au delà d'une certaine valeur, le nombre d'assignations initiales incorrectes est trop élevé pour être corrigé par l'approche hiérarchique ($\Delta\alpha$=27°).

Licence CC BY-NC-ND 3.0

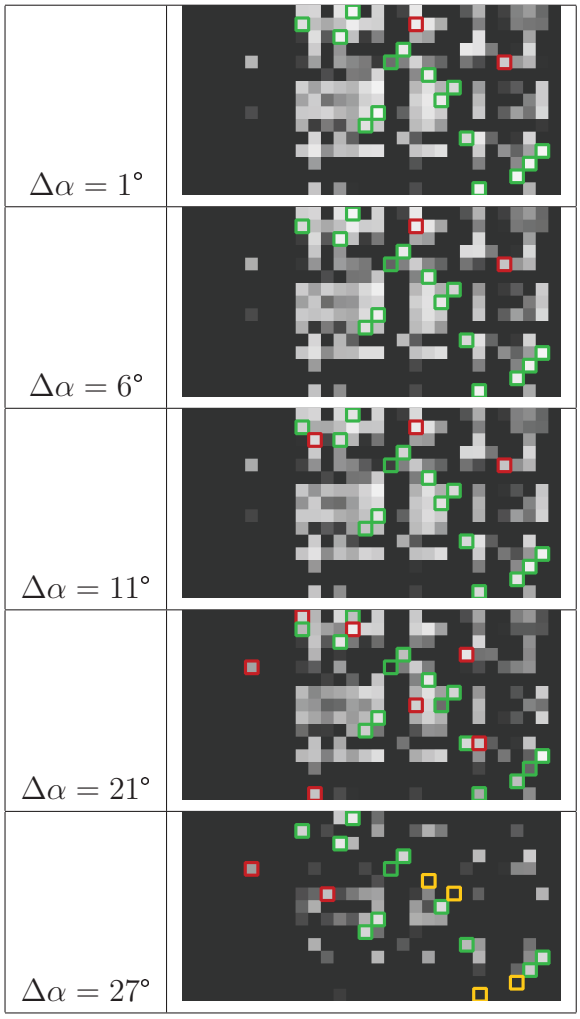


Figure 4.20. – Cartes de couts en fonction du pas d'échantillonnage de la rotation.

La figure 4.21 montre les valeurs de dissimilarité en fonction de l'angle α, entre le nœud 13 (G_1) et le nœud 26 (G_2) dans l'exemple du chapitre précédent (Figure 3.5). On peut voir l'effet de ce pas d'échantillonnage sur la mesure de dissimilarité dans la figure 4.22 qui montre la zone où la valeur de dissimilarité est inférieur au seuil C_{max} (Figure 4.22). Le cout d'assignation entre deux nœuds est la valeur de dissimilarité minimale obtenue pour toutes les valeurs de α testées. Dans l'exemple, la valeur minimale est obtenue pour $\alpha = 15°$. L'augmentation du pas $\Delta\alpha$ rend plus incertain l'obtention du minimum global. Dans cet exemple, les valeurs de dissimilarité inférieures au seuil s'étendent sur environ 20°. Un pas d'échantillonnage plus grand que cette étendue risque de donner une mesure de dissimilarité au dessus du seuil C_{max} alors que les nœuds sont similaires.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

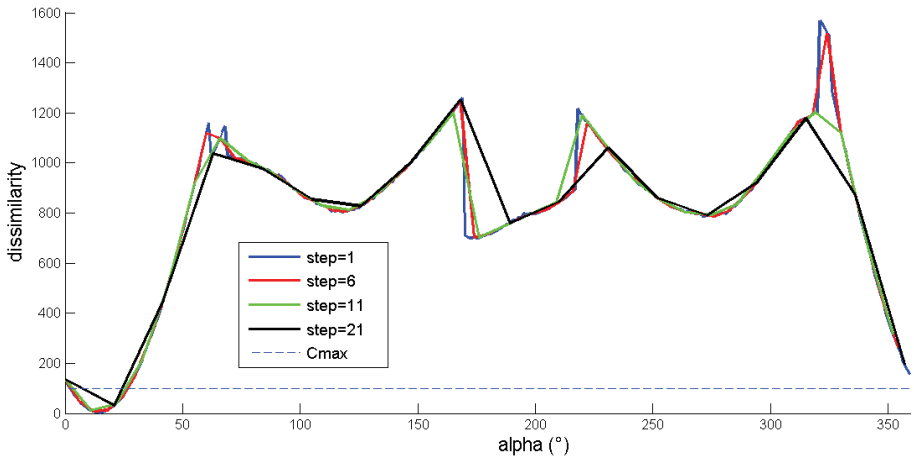


Figure 4.21. – Variation de la dissimilarité en fonction de l'angle α pour plusieurs valeurs de pas $\Delta\alpha$.

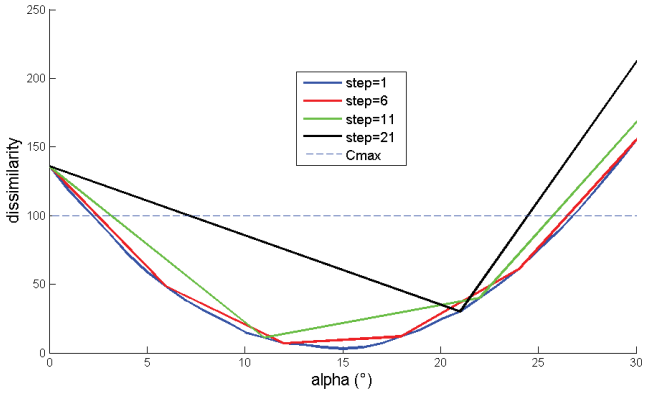


Figure 4.22. – Zoom sur la zone en dessous du seuil C_{max} de la figure 4.21.

Licence CC BY-NC-ND 3.0

Variation de la pondération de la différence entre les degrés

Lorsque les nœuds ont des degrés différents, leur forme est différente et ainsi la valeur de dissimilarité est supérieure à C_{max}, la valeur du paramètre de pondération affecte donc peu la mesure de dissimilarité. Néanmoins, une valeur élevée de cette pondération assure que deux nœuds avec des degrés différents ne seront pas assignés par la méthode Hongroise.

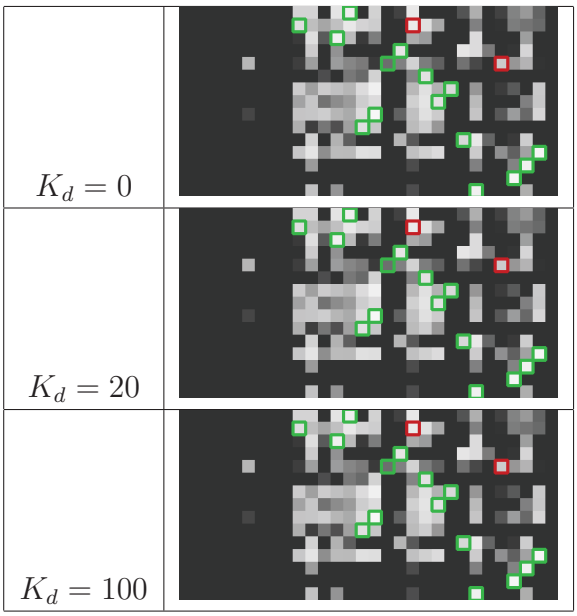


Figure 4.23. – Carte de couts en fonction de K_d. $K_d = 0$ signifie qu'on ne tient pas compte des degrés des nœuds dans la mesure.

Variation de la pondération de l'erreur quadratique

Ce paramètre pondère l'élément principal de la mesure de dissimilarité. Une valeur trop importante rend la mesure trop discriminante et l'algorithme ne fait pas assez d'assignations correctes pour obtenir un résultat correct au final. C'est le cas pour $K_c = \frac{1}{2}$ et $K_c = \frac{1}{3}$ (Figure 4.24), les assignations en orange n'ont pas été trouvées dans le résultat final, le matching est donc faux. On peut clairement voir l'impact de ce paramètre sur la matrice de couts : plus la carte de couts est « claire » moins la mesure est discriminante. Néanmoins, même pour des valeurs faibles ($K_c = \frac{1}{100}$), le résultat final est correct malgré l'augmentation du nombre d'assignations initiales incorrectes.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

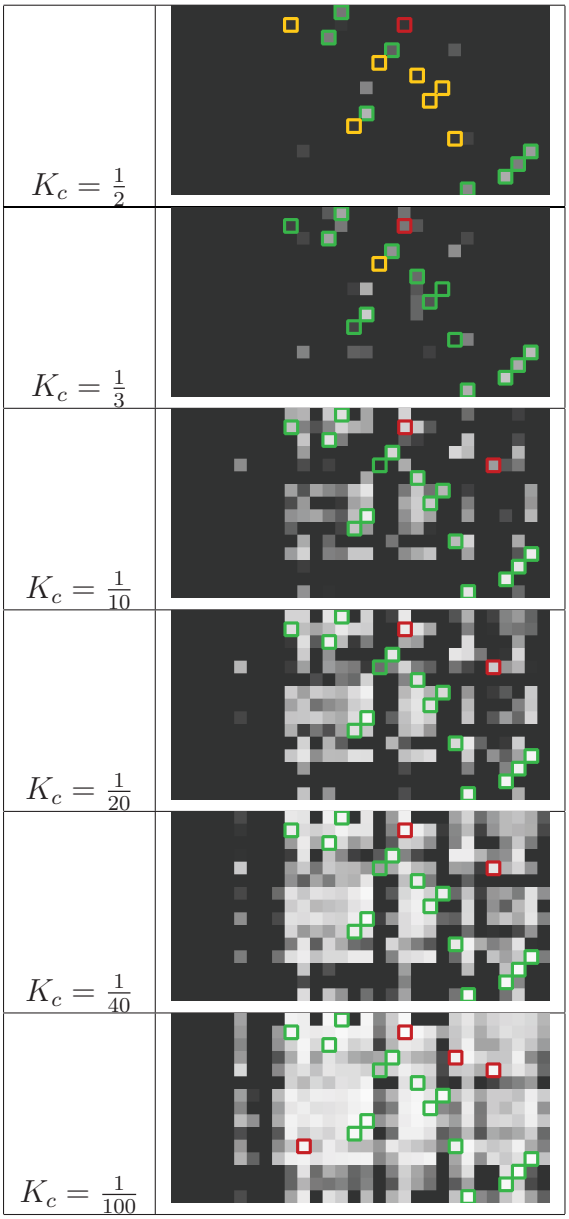


Figure 4.24. – Carte de couts en fonction de K_c.

Variation de la pondération de l'écart angulaire

Ce paramètre pondère la somme des différences entre les angles d'incidences entre les arêtes. Dans l'exemple de la figure 4.25, on peut voir que le nombre de fausses assignations est minimal pour $K_a = \frac{1}{10}$ ou $K_a = \frac{1}{20}$. Le nombre de fausses assignations augmente lorsque l'on s'éloigne de ces valeurs mais il ne produit pas de matching incorrect.

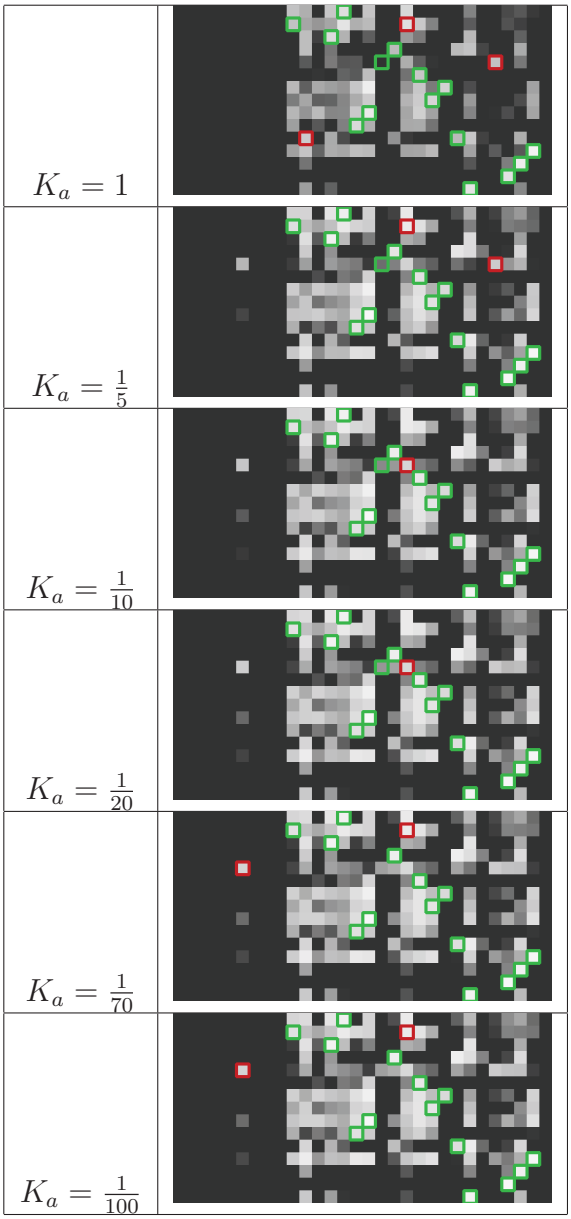


Figure 4.25. – Carte de couts en fonction de K_a.

Variation de la pondération de l'écart au facteur d'échelle moyen

Ce paramètre pondère le terme lié à l'écart au facteur d'échelle moyen obtenu après l'analyse statistique des assignations obtenues pour la mesure de dissimilarité initiale. Ce paramètre n'est présent que dans la mesure raffinée. Dans la figure 4.26, on peut noter que l'augmentation de cette pondération, augmente la discriminance de la mesure de dissimilarité raffinée. Au delà d'une certaine valeur (ici $K_\lambda = 100$), le nombre d'assignations possibles devient trop faible et le résultat n'est plus correct (il manque l'assignation en orange).

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

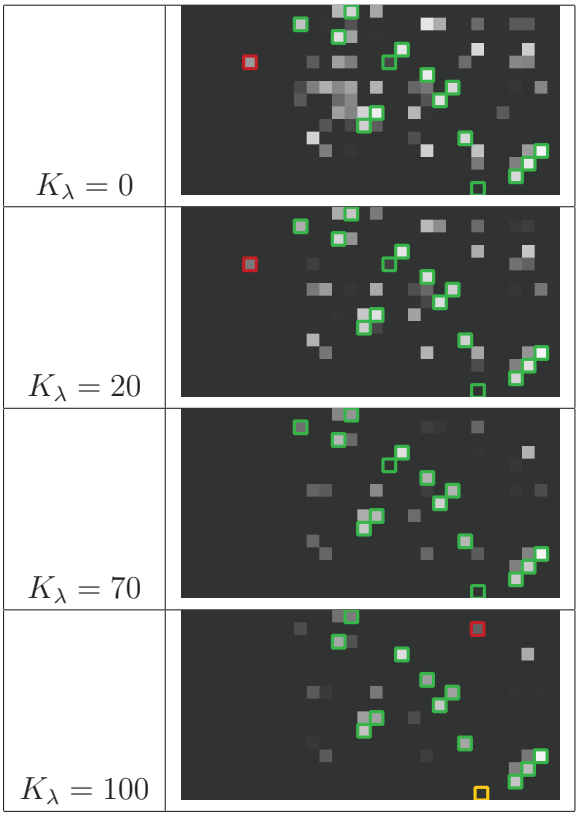


Figure 4.26. – Carte de couts en fonction de K_λ.

Variation de la pondération de l'écart à l'orientation moyenne

Ce paramètre pondère le terme lié à l'écart à l'orientation moyenne obtenue après l'analyse statistique des assignations obtenues pour la mesure de dissimilarité initiale. Ce paramètre n'est présent que dans la mesure raffinée. Le paramètre est inversement propositionnel à l'écart type σ_α pour augmenter l'importance de l'orientation lorsque le matching initial présente une forte cohérence angulaire. Dans la figure 4.27, on peut noter que l'augmentation de cette pondération augmente la discriminance de la mesure de dissimilarité raffinée. Un coefficient K_α trop élevé risque de ne pas faire suffisamment d'assignations pour que les étapes utilisant la topologie et le consensus de projection fonctionnent.

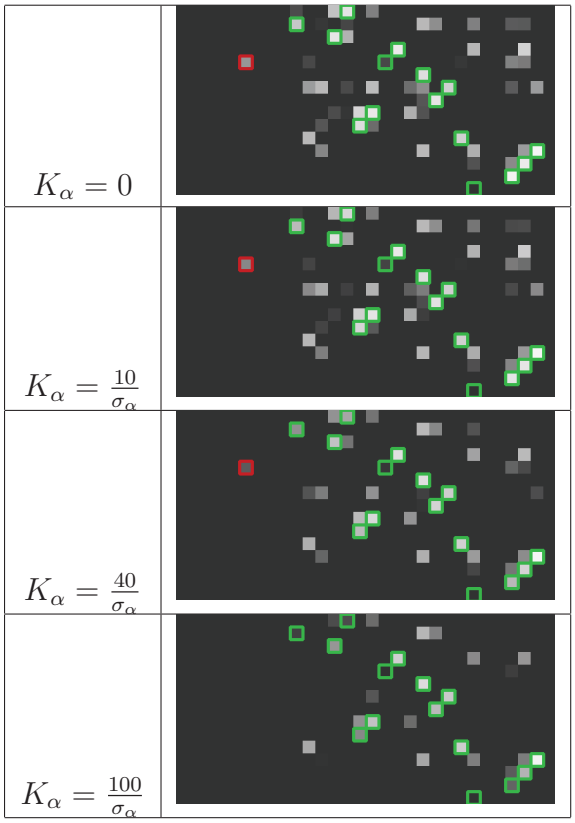


Figure 4.27. – Carte de couts en fonction de K_α.

Synthèse

Ces expériences ont été réalisées sur une série de matching et ont permis d'ajuster les paramètres utilisés dans les algorithmes et d'évaluer leur impacte sur la mesure de dissimilarité. Les paramètres présentés en table 4.1 ont été utilisés, sans modification, sur toutes les expérimentations. Notons que l'algorithme proposé n'a pas besoin de réglages précis pour donner un résultat correct et que ces paramètres sont transposables à d'autres environnements.

4.3.2. Évaluation de la mise en correspondance des cartes

Les premières expérimentations ont été faites sur l'environnement Pavin simulé. Le drone effectue une trajectoire à altitude fixe ou variable au dessus de la zone pour créer la carte globale. Le robot au sol suit une trajectoire définie à l'avance. Deux exemples de trajectoires sont présentées dans la figure suivante (Figure 4.28).

Cette expérience a permis de créer 11 cartes aériennes différentes et 13 cartes depuis le sol. Les cartes présentées dans le chapitre 2 et celles qui ont servi de fil-rouge dans le chapitre 3 proviennent de ces expérimentations. Les cartes aériennes diffèrent par

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

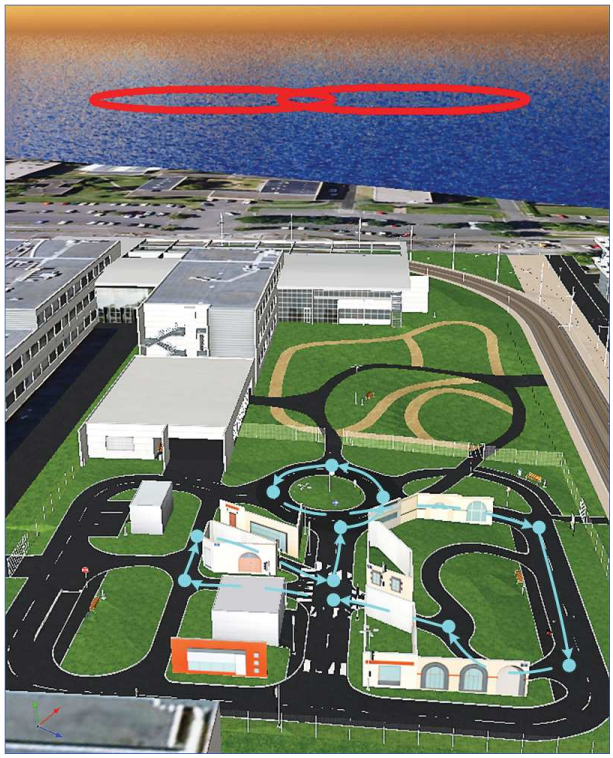


Figure 4.28. – Trajectoires effectuées par les robots dans PAVIN.

leur altitude, leur niveau de complétion au regard de la zone entière, mais également de la partie cartographiée (Est/Ouest) (Carte aérienne partielle (AP)). La méthode utilisée pour la cartographie varie également (image complète (AIM) et mosaïque de graphes (AGM)). Les cartes terrestres sont divisées en deux parcours. Un long parcours (en bleu clair dans la figure 4.28) correspond à 4 graphes qui diffèrent par les paramètres de segmentation et squelettisation utilisés lors de leur création (GPc). Un parcours plus restreint où 7 graphes correspondent aux cartes successives lors de la création des nœuds qui le composent ($PPc2 \rightarrow PPc8$). Ce parcours est également utilisé pour créer 2 cartes spéciales ayant des représentations différentes du carrefour central de Pavin ($SPc1$ et $SPc2$). Les tableaux suivants montrent un échantillon des résultats obtenus et le taux de chevauchement associé. Les lignes correspondent aux cartes aériennes et les colonnes aux cartes terrestres. Le premier tableau (Tableau 4.2) montre le nombre de nœuds mal assignés, et le deuxième (Tableau 4.3) montre le taux de chevauchement des matchings respectifs calculé comme suit :

$$\text{Taux de chevauchement} = \frac{N_{commun}}{|N_1| + |N_2| - N_{commun}} \tag{4.4}$$

Où $|N_1|$ et $|N_2|$ sont les dimensions des graphes G_1 et G_2, et N_{commun} est le nombre de noeuds en commun entre les deux graphes.

Licence CC BY-NC-ND 3.0

Erreur d'assignation	PPc2	PPc3	PPc4	PPc5	PPc6	PPc7	PPc8	SPc1	SPc2	GPc
AGM	0	0	0	0	0	0	0	0	0	0
AIM	0	0	1	1	0	0	0	0	1	0
AP	2	1	3	3	3	3	1	2	2	1

Table 4.2. – Nombre de nœuds mal assignés lors du matching.

Chevauchement (%)	PPc2	PPc3	PPc4	PPc5	PPc6	PPc7	PPc8	SPc1	SPc2	GPc
AGM	5	7	9	12	14	16	19	21	23	35
AIM	7	10	13	16	20	23	26	30	33	50
AP	0	4	8	13	12	12	11	11	14	20

Table 4.3. – Nombre de nœuds mal assignés lors du matching.

La vérité terrain est obtenue manuellement. À l'aide de la série de cartes successives ($PPc2 \rightarrow PPc8$), on peut voir que la réussite du matching est corrélée au nombre de nœuds et au pourcentage de chevauchement mais pas uniquement. Les résultats sont aussi liés à la différence entre les nœuds parcourus. Si plusieurs nœuds se ressemblent trop, le processus d'assignation peut assigner des faux positifs. Il suffit alors de cartographier un peu plus la zone pour trouver d'autres nœuds plus distinctifs.

Un autre problème est la différence de représentations d'une même intersection entre les deux cartes, comme abordé dans la discussion du chapitre précédent. L'intersection centrale de Pavin en est un bon exemple. Dans certaines cartes terrestres, elle est représentée par 2 nœuds (SPc1) et dans d'autres il y en a 3 (SPc2), (Figure 4.30). Il en va de même pour la carte aérienne, lorsque l'altitude d'observation augmente, la précision diminue et l'intersection peut être représentée de différentes façons. Les obstructions partielles peuvent également modifier la forme ou la topologie de l'intersection. Lorsque cette différence de représentation entre les deux cartes apparait, le matching est souvent erroné. En effet, l'approche proposée utilise un matching point à point et l'assignation de plusieurs nœuds à un seul n'est pas possible. Il peut cependant arriver que les deux représentations coexistent en même temps dans la carte aérienne (Figure 4.29). Dans ce cas, le processus d'appariement choisit automatiquement la représentation adaptée (Figure 4.31). Ces figures montrent deux fusions de cartes ayant la même carte aérienne, qui possèdent deux représentations de l'intersection centrale, et deux cartes terrestres qui ont une représentation différente de cette intersection. Les deux matchings sont corrects et les cartes fusionnées s'adaptent également.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

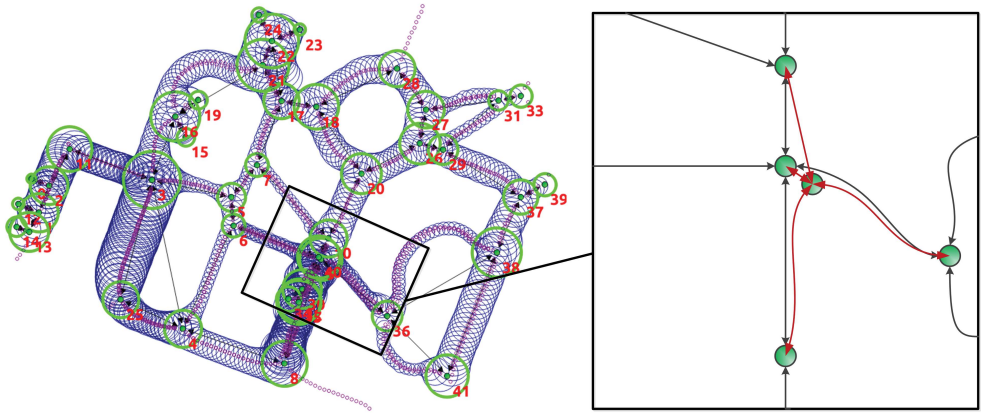


Figure 4.29. – Carte aérienne avec représentation multiple. (droite) Zoom sur la topologie de l'intersection centrale.

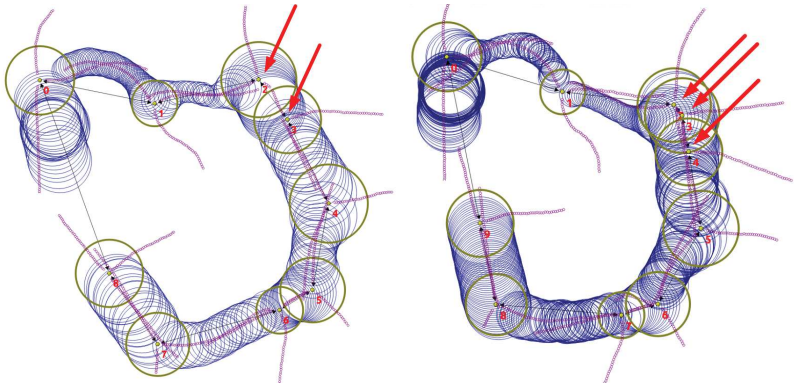


Figure 4.30. – Cartes terrestres avec des représentations différentes du même lieu. Les flèches rouges indiquent les nœuds présents dans l'intersection centrale.

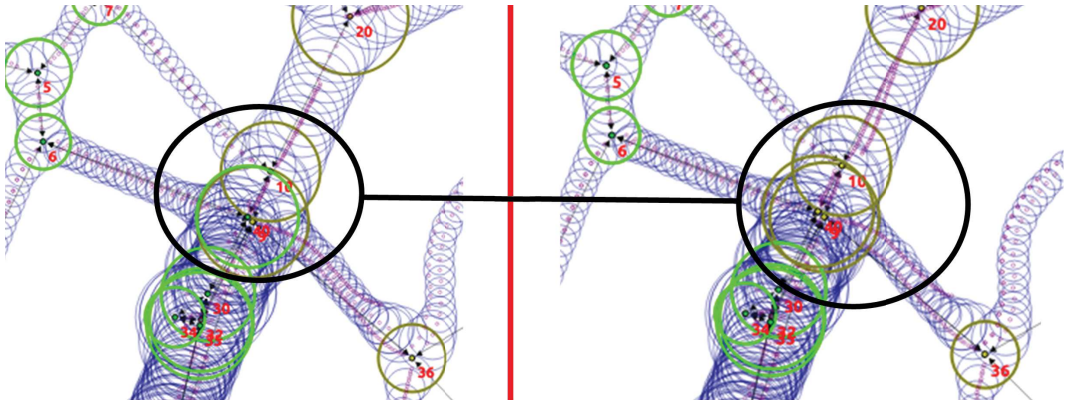


Figure 4.31. – Zoom sur l'intersection centrale après fusion des cartes avec des représentations multiples. Les cercles verts montrent les nœuds provenant de la carte aérienne et les cercles marrons ceux qui proviennent de la carte terrestre.

Les cartes ont été correctement matchées malgré leur différence de représentations. Il y a bien deux nœuds terrestres dans la carte fusionnée de gauche et trois dans celle de droite. Cette solution est possible car ces deux représentations coexistent dans la carte aérienne. Cependant, cette double représentation impacte la topologie des nœuds et également leur formes puisqu'ils ont plus d'arêtes que nécessaire. Néanmoins, l'utilisation des techniques d'assignation grâce au voisinage et par projection permettent de garantir un matching correct malgré cette différence de forme entre les nœuds.

4.3.3. Localisation à partir de la carte fusionnée

Cette deuxième série expérimentations est faite dans un environnement simplifié mais avec des robots et des capteurs réels. Cet environnement est construit pour que les nœuds ne se ressemblent pas. Le matching est ainsi facilité. La seule difficulté pour le matching vient de la ressemblance entre les nœuds 7 et 8 qui forment tous les deux un « T ». Mais leur orientation différente permet de les différencier avec la mesure raffinée. Plusieurs trajectoires ont été effectuées avec le robot, l'une d'entre elles est représentée (en violet) sur la carte fusionnée (Figure 4.32). On notera que les intersections qui ne sont pas visitées par le robot au sol ne sont pas mises à jour dans la carte fusionnée (cercles verts).

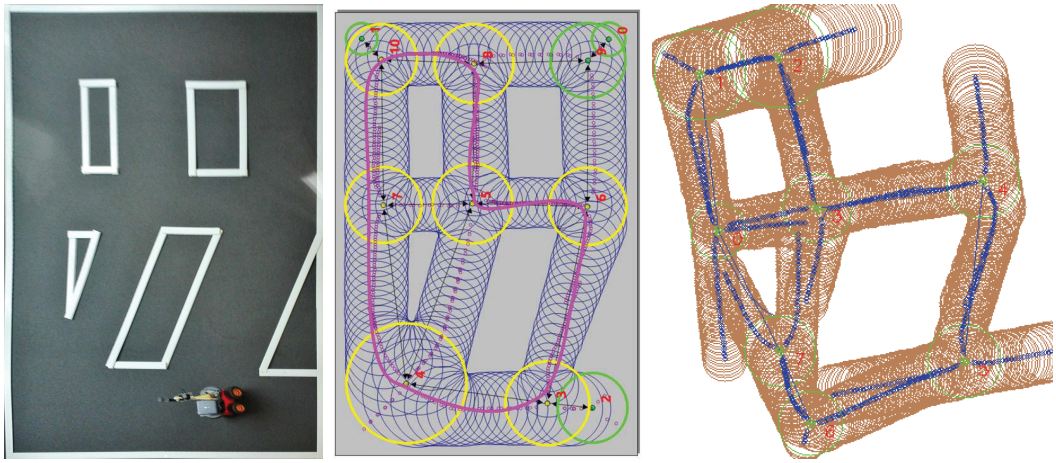


Figure 4.32. – Fusion des cartes aéro-terrestres dans un environnement intérieure. (gauche) Orthoimage de l'environnement, (milieu) carte augmentée, (droite) carte terrestre après l'exécution de la trajectoire en violet .

Dans cette série d'expériences, les séquences sont bouclées et enregistrées dans des fichiers *Rosbag* pour pouvoir être rejouées. Ces données enregistrées ont permis de faire varier le point de départ de la séquence et de tester les fermetures de boucles de la navigation terrestre.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Une fois les cartes fusionnées, nous avons testé la capacité du robot terrestre à se localiser dans cette nouvelle carte. Pour cela, le robot terrestre doit reconnaitre l'intersection la plus proche du lieu où il se trouve parmi les nœuds présents dans la carte, quelle que soit leur origine (terrestre ou aérienne). Pour faire cette reconnaissance de lieu, le robot terrestre utilise une mesure de dissimilarité semblable à la mesure de dissimilarité raffinée (Équation 3.5). Le facteur d'échelle moyen est alors égal à 1 et l'orientation moyenne est celle du robot. Une fois que le robot a détecté qu'il se trouvait proche d'un nœud existant à l'aide d'un seuil de validité, il utilise la transformation locale estimée à partir des informations de la mesure de dissimilarité pour se (re)localiser dans la carte globale. Cela permet de remettre à zéro la dérive de l'odométrie. Si le nœud n'existe pas dans la carte globale, le robot terrestre le rajoute et continue son exploration. Les trois cartes dans la Figure 4.33 montrent trois instants successifs lors de la re-localisation du robot (point noir). Dans les deux premières, on peut voir la dérive de la carte estimée par le robot grâce aux cercles verts qui ne se superposent pas aux routes en bleu. La deuxième image montre l'instant de la reconnaissance de l'intersection. Le petit trait noir représente l'erreur en position. Dans la dernière image, le robot est déplacé dans la carte pour réajuster sa position. On peut voir la réduction de la covariance de la position du robot dans la réduction du rayon du cercle gris entourant le robot.

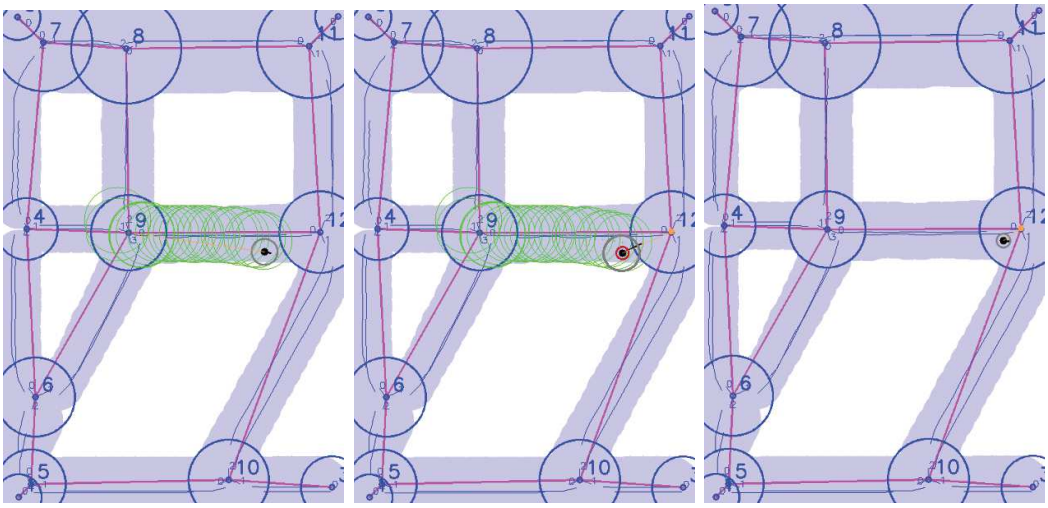


Figure 4.33. – Re-localisation du robot à l'aide de la carte augmentée. (gauche) La carte estimée par l'odométrie est représentée en vert (milieu) Le robot reconnait l'intersection qu'il perçoit comme étant un nœud de la carte (droite) Le robot se localise dans la carte augmentée grâce à cette reconnaissance de lieu.

D'autres tests ont été réalisés dans l'environnement Pavin simulé. Les résultats ont été concluants, le robot terrestre se localise correctement lorsqu'il passe par des intersections qu'il a déjà visitées. Lorsqu'il passe par des nœuds créés par le drone, soit il se relocalise correctement, soit il crée une nouvelle représentation de l'intersection qui correspond mieux à ses perceptions. Dans ce cas, deux représentations

Licence CC BY-NC-ND 3.0

de la même intersection coexistent. Les lieux n'étant pas équipés de système de localisation absolu, nous n'avons pas pu obtenir de résultats quantitatifs quant à la précision de la localisation. Seule la capacité du robot à se localiser a été évaluée.

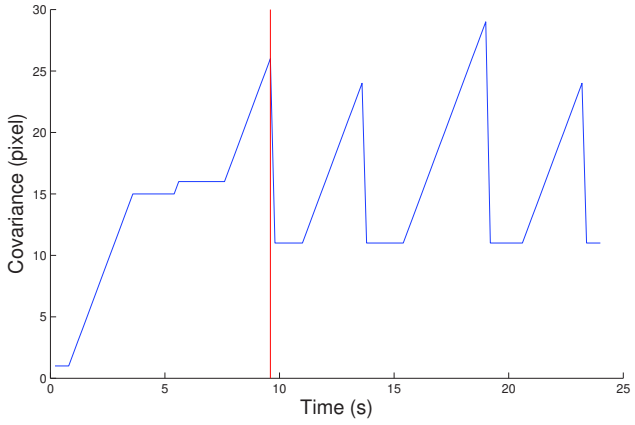


Figure 4.34. – Évolution de la covariance sur la localisation du robot

On peut voir sur la figure 4.34, la covariance sur le positionnement du robot, au cours du temps lors des expériences ayant donné lieu aux images précédentes (Figure 4.33). Le trait rouge indique le moment de la fusion des cartes aéro-terrestre. Les paliers marquent les endroits où le robot se localise par rapport à une intersection. Dans la première partie (à gauche du trait rouge), la covariance ne fait qu'augmenter car le robot crée de nouveaux nœuds sans se re-localiser. Après la fusion des cartes (à droite), le robot dispose de la carte globale et peut identifier sa position et ainsi réduire sa covariance. La dérive de la position du robot terrestre peut de nouveau augmenter mais est réduite à chaque nouvelle fermeture de boucle. Comme les nœuds avec lesquels il se re-localise proviennent du drone, la covariance est réduite, ici, à une valeur minimale de 10.

La cartographie coopérative n'est pas limitée à une seule fusion de cartes. Si le drone a effectué l'exploration d'une autre zone et revient communiquer avec le robot terrestre, le matching entre la carte augmenté et la nouvelle carte du drone est facilité puisqu'elles partagent une zone identique.

4.4. Utilisation de cartes en ligne

Les bases de données en ligne sont une autre source d'information qui peut être utilisée. Le projet OpenStreetMap met à disposition sa base de données gratuitement. Une carte en ligne est composée du réseau routier mais également des informations que les utilisateurs ont bien voulu y intégrer comme les bâtiments, les accès, les réseaux de bus, etc. Nous avons développé un programme pour exploiter la carte

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

d'une zone quelconque et la transformer en carte hybride. Elle peut alors être comparée avec la carte créée par le drone.

4.4.1. Création de cartes géoréférencées

Le site web *OpenStreetMap.org* propose une interface pour sélectionner la zone que l'on souhaite télécharger. Il offre également une interface pour les programmes (API) qui permet de faire une requête directement à la base de données pour obtenir les informations. Les paramètres à envoyer sont les limites de latitude et de longitude de la zone à télécharger.

Le fichier obtenu est un fichier XML avec l'extension *.osm*. Il contient une liste d'éléments hiérarchisés. D'abord, une liste de points géo-référencés puis une liste de routes constituées d'une liste de points et de différentes informations appelées « Tag ». Pour créer la carte hybride et la visualiser, nous projetons les informations géo-référencées dans le repère local tangent au point central de la zone téléchargée. Une image peut alors être créée en définissant une échelle pixel/mètre et en effectuant une interpolation entre les points successifs des routes que l'on souhaite afficher. Cette méthode dessine un "squelette" des routes dont on peut extraire le graphe pour construire la carte hybride. Le fichier OpenStreeMap ne fournit pas les informations sur la largeur des routes et des intersections. De plus, les points géo-référencés n'appartiennent pas obligatoirement au squelette de l'espace navigable (le milieu de la route). Néanmoins, la forme des routes créées est représentative de la forme réelle de ces routes. Les détails du processus de téléchargement, des fichiers OSM, du filtrage des routes avec les informations sémantiques et de la projection dans le repère local tangent sont disponibles en annexe A.

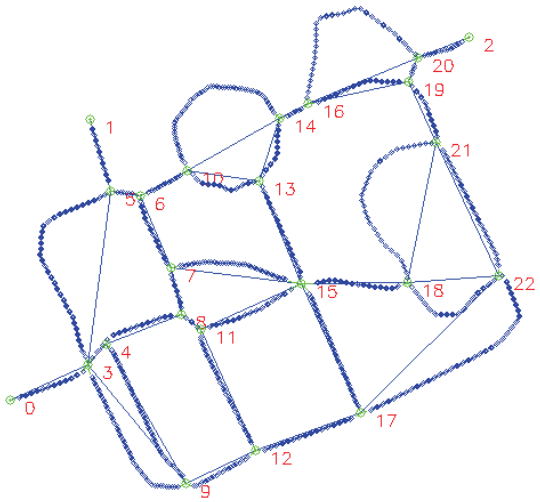


Figure 4.35. – Carte hybride du site Pavin construite avec OpenStreetMap.

La carte ci-dessus (Figure 4.35) montre le réseau routier obtenu pour l'environnement Pavin. On peut remarquer que le réseau n'est pas identique à celui présenté

précédemment (Figure 2.15). Il manque deux intersections à la partie haute du rond-point et la zone à gauche possède deux voies parallèles contrairement à la représentation aérienne qui n'en a qu'une. L'intersection centrale est représentée par un seul et unique point.

Un autre changement important est le fait que le tracé ne suit pas le squelette de la route. Ainsi, la forme des arêtes ne correspond pas parfaitement à celle de la carte aérienne. Le graphe étant construit sur le squelette de l'espace navigable, les arêtes sont légèrement courbées au niveau des nœuds. Le schéma suivant (Figure 4.36) montre un exemple de ce phénomène. Dans une intersection en « T », les angles entres les arêtes sont de 90, 90 et 180 degrés dans la représentation OSM (en violet) alors que dans l'axe médian (en bleu), la répartition angulaire tend vers 120,120,120. Cette différence crée un décalage entre les branches lors de la mesure de dissimilarité puisqu'elle est calculée en superposant les centres.

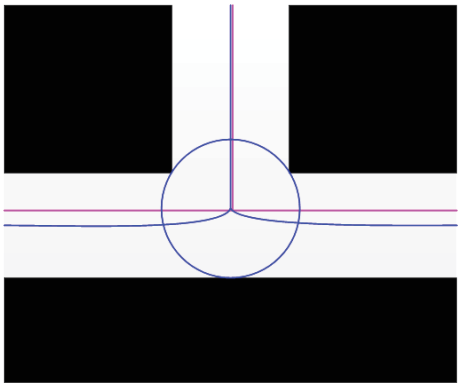


Figure 4.36. – Différence de représentations entre la carte OSM (en violet) et représentation par axe médian (en bleu).

4.4.2. Matching des cartes OSM et aérienne

Pour fusionner la carte provenant de OpenStreetMap avec la carte aérienne construite par le drone, nous utilisons la même approche que pour le matching aéro-terrestre. La mesure de dissimilarité est quasiment la même, seul le calcul du facteur d'échelle change. En effet, la carte OSM ne possédant pas d'information sur le rayon de l'intersection, il faut utiliser un autre élément. Nous avons choisi d'utiliser le rapport des longueurs de l'arête la plus courte comme facteur d'échelle. Cette information est disponible dans les deux cartes et représente bien le facteur d'échelle si les formes des arêtes correspondent.

Ensuite, une détection des erreurs d'assignation est faite en utilisant la projection globale calculée avec la méthode RANSAC, comme pour la mosaïque de graphes. Cela permet également d'assigner d'autres nœuds grâce à leur position. Un autre avantage du calcul de la transformation rigide est que nous pouvons l'utiliser pour

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

déterminer si le matching est correct ou non. Si les déformations perspectives présentes dans la transformation sont trop importantes, cela signifie que les nœuds sont mal assignés.

La figure 4.37 montre le résultat du matching entre la carte OSM présentée plus haut et la carte aérienne. Les traits bleu clair montrent les associations initiales en utilisant la mesure de dissimilarité et la méthode Hongroise. Les erreurs détectées par la méthode RANSAC sont indiquées en rouge et les assignations obtenues par projection sont en bleu foncé. Les arêtes en orange indiquent le matching implicite des arêtes.

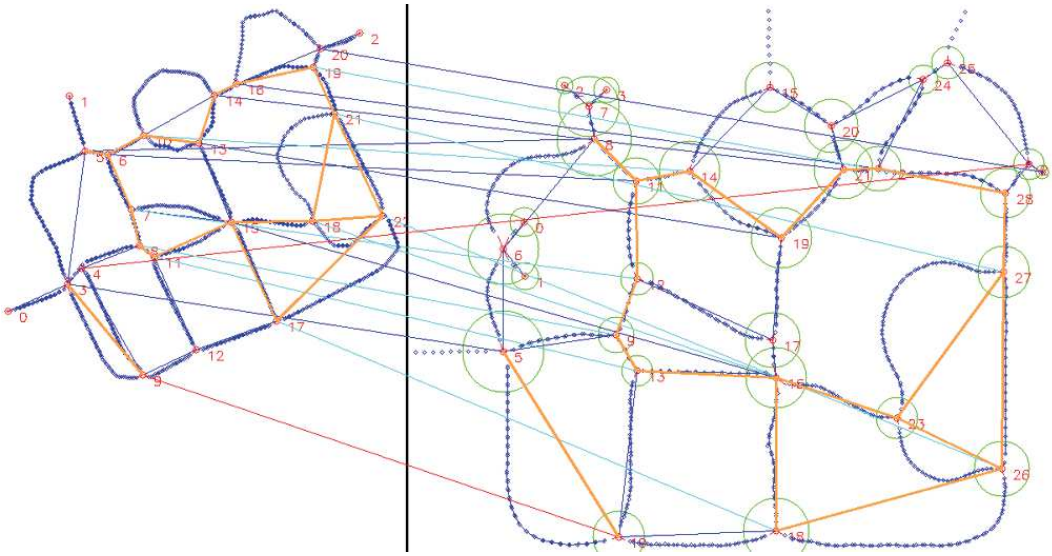


Figure 4.37. – Matching entre la carte routière et la carte créée par le drone.

Ce matching a été testé avec plusieurs cartes aériennes du site Pavin. Le matching n'est jamais parfait puisque la représentation topologique est différente. Cependant, on peut distinguer les matchings ayant réussi à associer les nœuds correctement là où il n'y a pas d'ambigüité et ceux qui ont complètement échoué. Sur tous les tests, seules les cartes aériennes avec une représentation complète de l'environnement ont réussi. Lorsque la cartographie n'est pas complète, les arêtes qui ne sont pas encore explorées faussent le facteur d'échelle et donc la comparaison de la forme. Il en va de même avec les représentations multiples qui peuvent faire apparaitre de petites arêtes qui faussent le facteur d'échelle. Dans le cas de Pavin, le changement de topologie affecte beaucoup le matching. Sur d'autres environnements, avec une topologie plus stable, les résultats pourraient être meilleurs. Si elles sont disponibles, les données GPS acquises par le drone peuvent servir au matching pour calculer un facteur d'échelle plus précis et utiliser la position des nœuds lors du calcul de dissimilarité.

Nous pouvons aussi utiliser le matching, s'il est réussi, pour fournir l'information des coordonnées GPS aux nœuds de la carte aérienne. Pour cela, en faisant l'hypothèse du monde plan, on peut utiliser une interpolation linéaire des coordonnées GPS

entre deux nœuds matchés. Cette approximation est suffisante si l'on n'a pas besoin d'une localisation très précise. L'image suivante (Figure 4.38) montre la superposition des cartes en utilisant la transformation globale obtenue grâce au matching. La grille montre les coordonnées de longitude et de latitude. Quelques coordonnées sont affichées pour montrer que les informations correspondent. La fusion n'est pas utilisée pour corriger les déformations présentes dans la carte aérienne car, comme expliqué précédemment, les nœuds ne sont pas définis au même endroit.

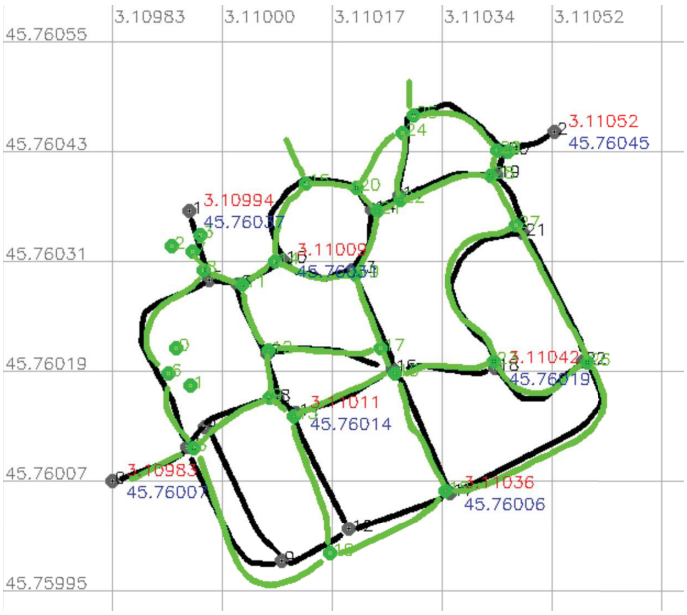


Figure 4.38. – Projection de la carte aérienne dans le repère de la carte OSM.

Ce matching ne peut pas être fait directement avec une carte terrestre. Premièrement, la normalisation par l'arête la plus courte nécessite d'avoir exploré toutes les arêtes d'un nœud pour être valide. Il faut donc trouver une autre solution pour déterminer le facteur d'échelle. Deuxièmement, la détection des associations erronées est faite en calculant une projection globale sous forme de transformation rigide. A cause des déformations de la carte terrestre, dues à l'odométrie, cette transformation rigide globale n'existe pas. On pourrait utiliser l'approche locale comme pour le matching aéro-terrestre mais le nombre de fausses associations initiales rend la méthode inutilisable en pratique. On ne peut donc pas faire de matching direct entre les cartes OSM et terrestre, mais on peut utiliser la carte aérienne comme intermédiaire. En fusionnant une carte aérienne et OSM, on obtient une carte aérienne augmentée avec des coordonnées GPS. On peut alors la fusionner avec une carte terrestre pour associer toutes les informations. Notons que la disponibilité de donnés de positionnement absolue type GPS sur le robot terrestre permettrait de fusionner les cartes OSM et terrestre directement sans passer par la carte aérienne.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

Discussion

Dans ce chapitre, nous avons présenté les expérimentations mises en place pour évaluer toutes les étapes de notre approche de cartographie coopérative aéro-terrestre.

Nous avons pu déterminer les avantages et les limites des cartes topologiques en milieu peu structuré. Ces limites sont en partie liées à la capacité à détecter précisément et de manière fiable l'espace navigable.

Nous avons proposé une solution pour gérer les éléments dynamiques qui peuvent altérer la représentation globale de la carte. Pour cela, une information de confiance est donnée aux éléments de la carte pour supprimer les nœuds éphémères et faire une sélection parmi des représentations multiples.

Le matching de cartes a été évalué sur plusieurs environnements virtuels et réels, avec des chevauchements et des tailles différentes. Les matchings réussis mènent à la création de cartes augmentées. Nous avons pu montré que ces cartes augmentées peuvent être utilisées pour la localisation et la navigation terrestre dans le sens où les fermetures de boucles sont possibles même avec les nœuds créés par la carte aérienne.

Cette approche aéro-terrestre a été étendue à l'utilisation de cartes provenant d'une base de données en ligne, OpenStreetMap. Le matching est plus complexe car les représentations sont différentes et ne possèdent pas les mêmes informations. Lorsque le matching est réussi, la carte aérienne peut être projetée dans le repère local tangent de la zone et l'on peut attribuer des coordonnées GPS aux nœuds de la carte.

Au cours de ces expérimentations, nous avons pu observer plusieurs phénomènes. Premièrement, le matching doit être suffisamment bon pour que la carte fusionnée soit cohérente. Des nœuds mal assignés entrainent des erreurs de cartographie. C'est pour cela que le critère permettant de déterminer si un matching est correct ou non est très important. Deuxièmement, la carte construite est utilisable par le robot terrestre, à différents niveaux. Nous avons montré la fermeture de boucle, et la relocalisation du robot, ainsi que l'intégration de nouveaux éléments pas encore visités.

De même, la cartographie aérienne augmentée d'informations terrestres peut servir pour définir des stratégies de navigation coopérative, définir des zones d'intérêt pour l'exploration et le survol.

Licence CC BY-NC-ND 3.0

5. Conclusion et perspectives

5.1. Bilan Général

Cadre de l'étude

Dans cette thèse nous avons cherché à fournir une perception augmentée et un référentiel commun à un système hétérogène aéro-terrestre. Nous avons, plus particulièrement, proposé une solution au problème de la cartographie coopérative de l'espace navigable terrestre. Le but était d'enrichir des cartes construites par des systèmes hétérogènes en les augmentant par des données indisponibles sans coopération. Nous avons, en effet, corrigé et augmenté la carte d'un robot terrestre grâce à des observations aériennes. Les drones et les robots terrestres ont des capacités de déplacement et de perception complémentaires qui, ensemble, permettent de mener à bien des missions complexes. Ceci explique l'essor important des systèmes collaboratifs aéro-terrestres de nos jours. Cependant, les points de vue de ces robots sont très différents et rendent difficile cette coopération. En effet, les méthodes classiques de SLAM dépendent d'amers fortement perturbés par le changement de points de vue ou sont difficilement utilisables pour la planification de chemins.

Représentation de l'environnement

Pour pallier ce problème, nous avons défini une représentation hybride de l'environnement pour qu'elle soit utilisable par un robot terrestre tout en étant constructible également par un drone. Ce modèle de l'environnement est une carte représentant la topologie de l'espace navigable terrestre à l'aide d'un graphe, dont les nœuds sont les intersections, et les arêtes sont les routes entre ces intersections. Les éléments de ce graphe sont enrichis de données métriques telles que la taille des intersections et la forme des routes, des donnés sémantiques peuvent y être ajoutées comme illustré dans certaines situations. Cette représentation est plus légère en mémoire qu'une représentation métrique dense et est plus riche qu'une carte topologique pure. Les méthodes de création de ces cartes ont été présentées pour des images perspectives prises par un drone, des images omnidirectionnelles catadioptriques acquises par un robot terrestre, et grâce à des cartes routières en ligne.

Fusion des cartes aéro-terrestres

Les cartes créées sont fusionnées grâce à une mise en correspondance de leurs éléments sur la base d'un appariement point à point. Cette mise en correspondance est une optimisation du cout global d'assignation des points. Le cout d'assignation entre les points est calculé à partir d'une mesure de dissimilarité adaptée aux informations discriminantes. Cette mesure de dissimilarité est construite pour être invariante aux facteurs inconnus, que sont les changements de repère et d'échelle. Le matching est ensuite filtré à plusieurs reprises en utilisant des approches statistiques, topologiques et métriques. Le résultat est finalement validé ou invalidé à l'aide d'un score calculé grâce au nombre de nœuds et d'arêtes matchés. La fusion des informations crée une carte augmentée en utilisant la structure globale aérienne qui est plus cohérente, mais avec des informations locales terrestres plus précises.

Expérimentations

Plusieurs expérimentations ont été faites en simulation et en environnement réel pour valider l'approche proposée et en déterminer les limites. Les résultats ont montré que l'approche proposée fonctionne dans les environnements structurés où l'espace navigable peut être détecté de manière fiable. La carte augmentée peut être utilisée directement par le robot terrestre pour la localisation et la détection de fermeture de boucle, pour la navigation et la planification de trajectoire. Il est même possible d'étendre cette approche à des cartes provenant de bases de données géographiques pour propager des données de géolocalisation.

5.2. Pour aller plus loin

Plusieurs idées ont été envisagées pour améliorer les différentes étapes de l'approche proposée.

La carte hybride créée dans cette étude est construite à partir d'images dont sont extraits l'espace navigable et son squelette. Aucune autre information n'est tirée de ces images car les points de vue différents rendent cette information peu utilisable dans le cadre aéro-terrestre. Il peut être intéressant d'ajouter les perceptions visuelles locales aux nœuds pour un matching plus robuste. Ce type d'information peut également faciliter la compréhension de la carte pour un utilisateur humain. L'ajout d'informations à la carte se fait cependant au prix d'une augmentation du volume mémoire.

Certaines informations sémantiques sont déjà présentes dans la représentation proposée. Premièrement, tous les éléments de la carte disposent d'une information sur l'origine de cet élément (drone, rover ou osm). Deuxièmement, d'autres informations sémantiques peuvent être extraites lors de la création des cartes. Dans le cas

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

des cartes provenant d'OpenStreetMap, des informations sont parfois disponibles directement dans le fichier. Dans le cas d'autres cartes, il est possible d'obtenir des informations à l'aide d'une classification en utilisant les images ou les données métriques, comme présenté dans la discussion du chapitre 2 (section 2.3.2). L'étude de ces capacités de classification a été limitée dans cette étude. L'utilisation de ce type d'information peut être un réel avantage que ce soit pour la création de cartes dynamiques, un matching plus robuste ou une planification contextuelle des trajectoires.

Cette approche peut également être utilisée pour la détection de changements dans l'environnement au cours du temps et la maintenance des cartes.

La fusion des cartes n'a été testé que dans le cadre d'une coopération entre un drone et un robot terrestre. Ces travaux peuvent être étendus à la coordination d'une flotte de robots hétérogènes comprenant plus d'agents.

Licence CC BY-NC-ND 3.0

A. Téléchargement des cartes OpenStreetMap

Le logigramme suivant montre les étapes de la création de carte qui seront détaillées dans cette annexe : L'interface avec le site, le fichier osm, ses informations et la projection dans le plan tangent local.

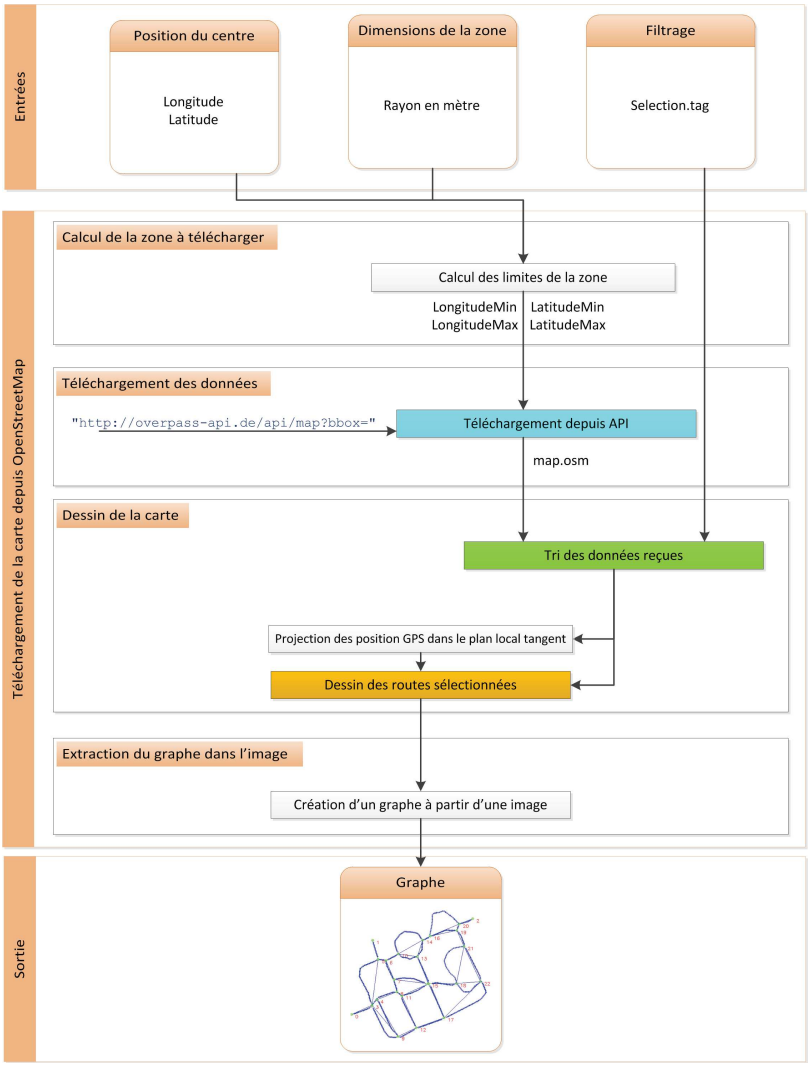


Figure A.1. – Logigramme du téléchargement de la carte OpenStreetMap.

A.1. API

Le site web OpenStreetMap fournit une interface pour exporter les informations d'une zone dans un fichier *map.osm*. Cette interface a besoin des limites en longitude et latitude pour générer le fichier.

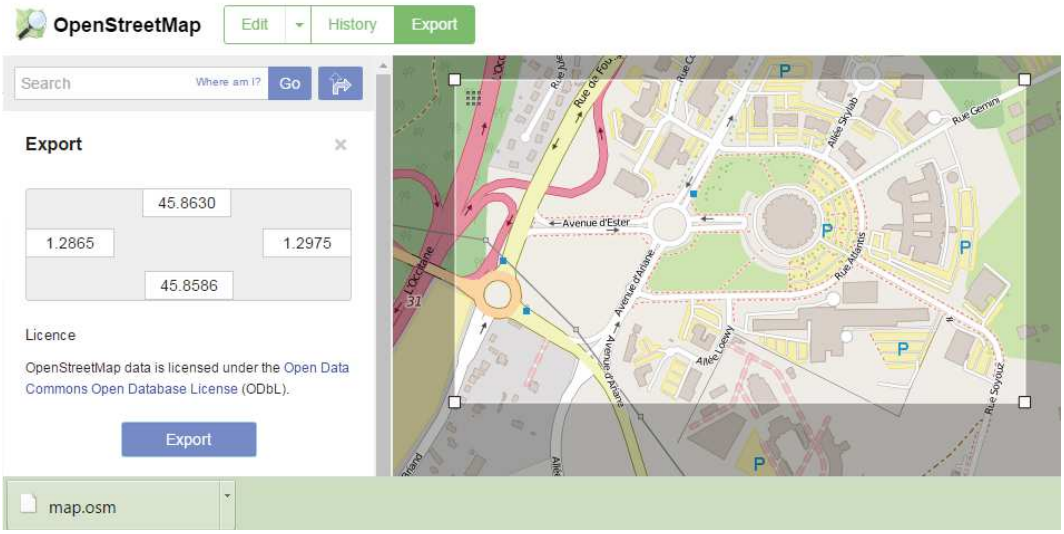


Figure A.2. – Interface web permettant de sélectionner la zone à télécharger.

Ces limites sont ensuite utilisées pour faire une requête à la base de données à l'aide du lien suivant :

$$http\ ://overpass\text{-}api.de/api/map\ ?bbox= \\ \ll longitudeMin \gg\ ,\ \ll latitudeMin \gg\ ,\ \ll longitudeMax \gg\ ,\ \ll latitudeMax \gg$$

Ce lien peut être utilisé directement depuis une application sans passer par le site. Nous avons développé un programme pour calculer les limites de la zone à exporter à partir des coordonnées GPS du centre et de la taille de cette zone. Le programme envoie ensuite la requête grâce au lien précédant et récupère le fichier.

A.2. Fichier .osm

Le fichier *map.osm* est un fichier au format XML. C'est un format à balise très utilisé. La figure suivante présente l'entête de ce fichier montrant l'origine des informations.

Ce format de fichier est constitué de points, de routes et de zones.

Les points « *node* » sont identifiés par un nombre « *id* » et possèdent une position géo-référencée : latitude et longitude mais pas d'altitude. Ces point peuvent, ou non,

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

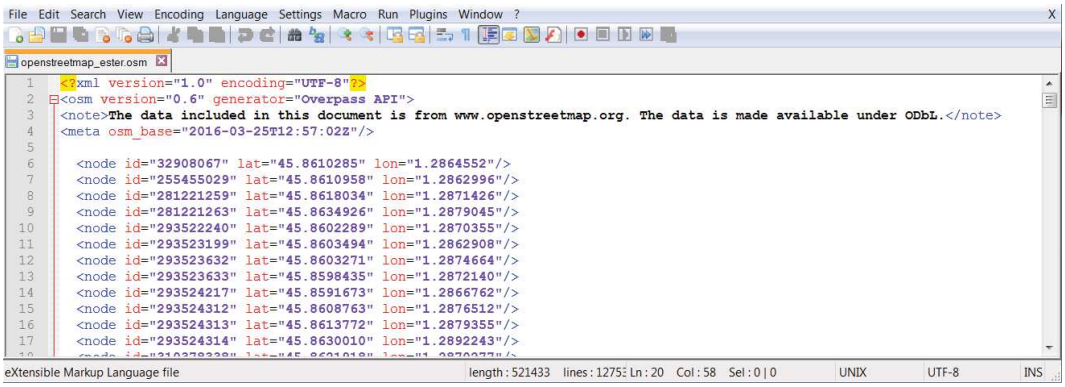


Figure A.3. – Format des fichiers .osm.

avoir des informations sémantiques « *tag* » définies par une clé « *k* » et une valeur « *v* ». Ces informations définissent souvent la position d'un objet particulier comme un arbre ou une boite aux lettres (Figure A.4).

```
<node id="651980068" lat="45.8612951" lon="1.2903966">
  <tag k="amenity" v="post_box"/>
  <tag k="operator" v="La Poste"/>
  <tag k="source" v="Tracking GPS"/>
</node>

<node id="2218381964" lat="45.8614570" lon="1.2937654">
  <tag k="natural" v="tree"/>
</node>
```

Figure A.4. – Exemple de points particuliers dans le fichier OSM.

Les routes « *way* » sont également identifiées par un numéro et sont constituées d'une liste de points « *nd* » dont la référence « *ref* » correspond à l'identifiant des points précédents. Les routes disposent également d'informations sémantiques qui seront très utiles pour le filtrage. On peut y voir le nom, le sens et le type de voie. Cependant, ces informations ne sont pas toujours renseignées.

```
<way id="26761692">
  <nd ref="390469555"/>
  <nd ref="2218381706"/>
  <nd ref="572504582"/>
  <nd ref="293524217"/>
  <nd ref="572504584"/>
  <nd ref="2218381701"/>
  <nd ref="293523633"/>
  <tag k="highway" v="tertiary"/>
  <tag k="name" v="Rue Aristide Briand"/>
  <tag k="oneway" v="yes"/>
  <tag k="ref" v="D 142"/>
</way>
```

Figure A.5. – Exemple d'une route dans le fichier OSM.

Enfin, les zones « *relation* » sont également identifiées par un numéro et sont constituées

Licence CC BY-NC-ND 3.0

d'une liste d'éléments « *member* ». Ces éléments peuvent être de différents types précisés dans la liste. Ainsi, une liste de points peut définir un polygone représentant un bâtiment ou une délimitation routière comme ci-dessous.

```
<relation id="4716475">
  <member type="node" ref="2218381955" role=""/>
  <member type="node" ref="3406176265" role=""/>
  <member type="node" ref="2218381967" role=""/>
  <member type="node" ref="3406176266" role=""/>
  <tag k="name" v="Ester Limoges Technopole"/>
  <tag k="operator" v="STCL"/>
  <tag k="public_transport" v="stop_area"/>
  <tag k="type" v="public_transport"/>
</relation>
```

Figure A.6. – Exemple d'une zone dans le fichier OSM.

Pour une vision simple et rapide, ce fichier peut être ouvert avec un logiciel de SIG tel que *QGIS*.

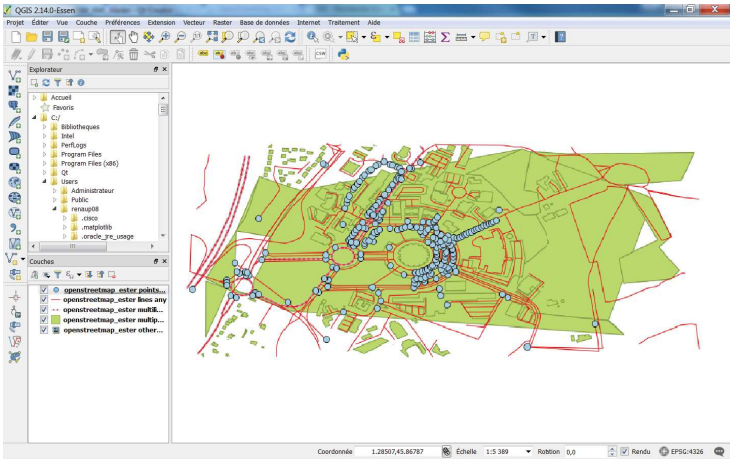


Figure A.7. – Fichier .osm ouvert à l'aide d'un logiciel de GIS.

On peut y voir les différents éléments présents dans le fichier : les points, les lignes, les zones. Les éléments du fichier peuvent être manipulés pour visualiser les informations qu'ils contiennent ou obtenir un rendu visuel plus compréhensible. En particulier, les attributs des routes peuvent être visualisés dans un tableau (Figure A.8).

La quantité d'information est très élevée et n'est pas forcément adaptée à l'application souhaitée.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

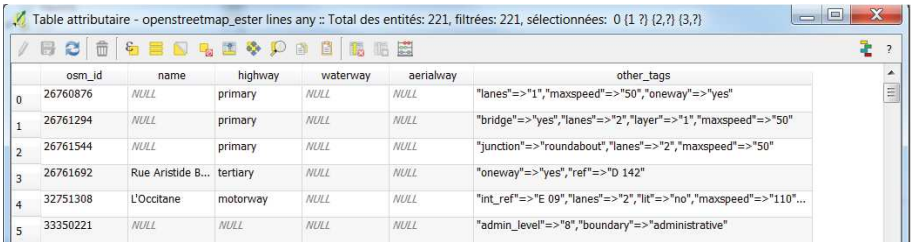


Figure A.8. – Informations portées par les routes dans OpenStreetMap.

A.3. Filtrage

Parmi les routes proposées, il peut se trouver des trottoirs, des lignes de bus, des chemins de particuliers, etc. Les informations du type de route peuvent être utilisées pour filtrer les routes que l'on souhaite utiliser.

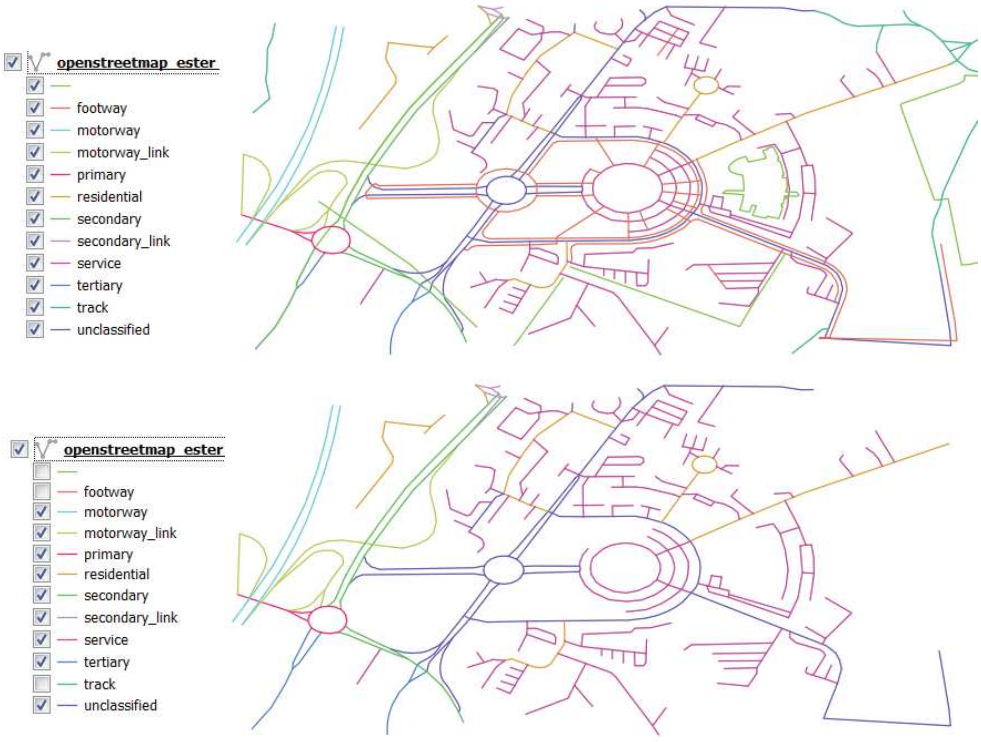


Figure A.9. – Carte avec différenciation du type des routes et filtrage.

A.4. Projection dans le repère local tangent

Les cartes sont géo-référencées. Il est nécessaire de projeter les informations de position pour obtenir des données métriques. Pour réaliser la carte, nous utilisons

Licence CC BY-NC-ND 3.0

une projection des coordonnées GPS dans le repère local tangent « East North Up » (ENU).

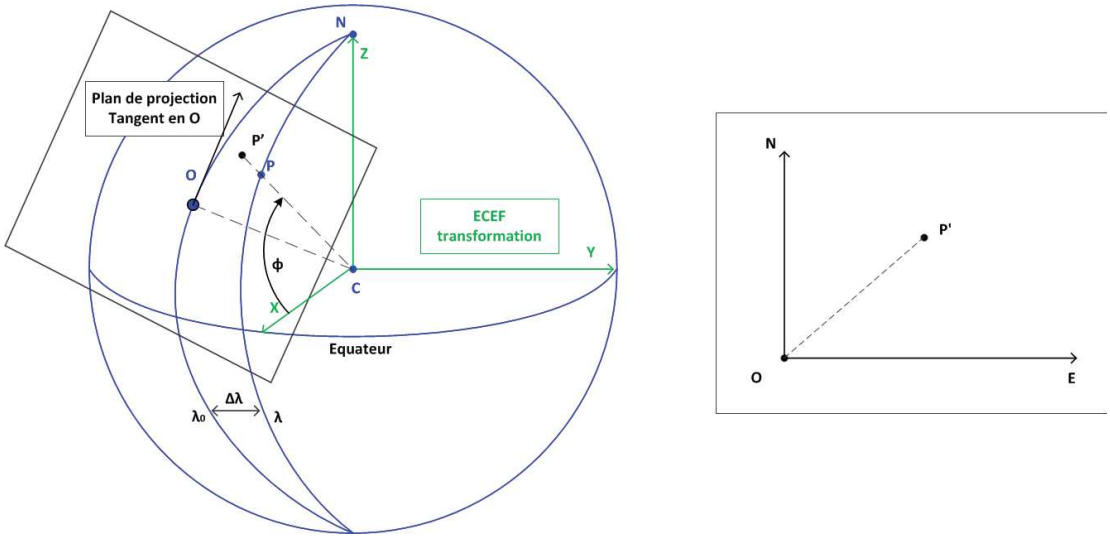


Figure A.10. – Changement vers le repère local tangent.

On réalise d'abord une transformation des coordonnées GPS en coordonnées cartésiennes centrées sur la Terre (ECEF). On obtient ainsi des coordonnées métriques par rapport au centre de la terre avec les formules suivantes :

$$\begin{cases} X = \frac{a}{\sqrt{1-e^2 sin^2\phi}} cos\phi\, cos\lambda \\ Y = \frac{a}{\sqrt{1-e^2 sin^2\phi}} cos\phi\, sin\lambda \\ Z = \frac{a(1-e^2)}{\sqrt{1-e^2 sin^2\phi}} sin\phi \end{cases}$$

$$\text{avec} \begin{cases} a = rayon\ de\ la\ Terre = 6378137m \\ e^2 = excentricité = 0.00669438 \end{cases}$$

La projection dans le repère local tangent, aussi appelée repère ENU, est calculée grâce aux coordonnées précédentes pour le point à déterminer et le point O, origine du repère :

$$P_{ENU} = R_O^T(P_{ECEF} - O_{ECEF})$$

Avec R_O^T la matrice de passage du repère centré sur la Terre vers le repère local tangent.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

$$R_O^T = \begin{bmatrix} -sin\phi & cos\phi & 0 \\ -sin\lambda cos\phi & -sin\lambda sin\phi & cos\lambda \\ cos\lambda cos\phi & cos\lambda sin\phi & sin\lambda \end{bmatrix}$$

Les positions GPS sont ensuite projetées dans une image en utilisant la résolution pixel/mètre que l'on souhaite. Les coordonnées sont des projections dans le plan local tangent, donc l'orthogonalité n'est valable que localement. Les routes sont obtenues par interpolation linéaire entre les projections des points qui la composent. La carte est ainsi redressée et les rond-points sont « rond ». Les routes ne contiennent pas d'information sur leur largeur, elle sont donc toutes tracées avec la même épaisseur de trait. L'image binaire obtenue (Figure A.11) peut alors être utilisée pour la création de la carte hybride.

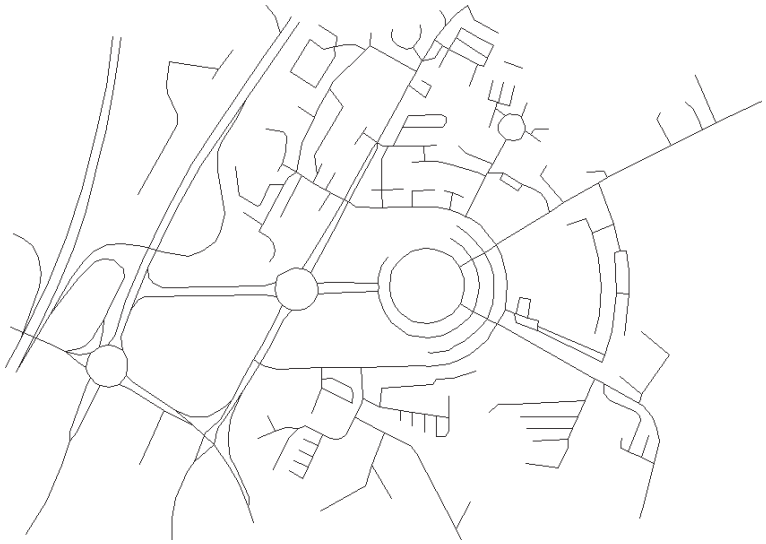


Figure A.11. – Carte représentée dans le repère local tangent.

Bibliographie

[Abdulkader, 1998] Abdulkader, A. M. (1998). *Parallel Algorithms for Labelled Graph Matching*. PhD thesis.

[Adriana Tapus, 2005] Adriana Tapus (2005). *Topological SLAM - Simultaneous Localization and Mapping with Fingerprints of Places*. PhD thesis.

[Almasi et al., 2018] Almasi, S., Lauric, A., Malek, A., and Miller, E. L. (2018). Cerebrovascular network registration via an efficient attributed graph matching technique. *Medical Image Analysis*, 46 :118–129.

[Atanasov et al., 2018] Atanasov, N., Bowman, S. L., Daniilidis, K., and Pappas, G. J. (2018). A Unifying View of Geometry, Semantics, and Data Association in SLAM. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

[Athanasios Mademlis, 2010] Athanasios Mademlis (2010). A Map-Search Framework Based on Attributed Graph Matching. *IEEE computer society*.

[Bazeille and Filliat, 2011] Bazeille, S. and Filliat, D. (2011). Incremental topometric slam using vision and robot odometry. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4067–4073. IEEE.

[Bechon, 2016] Bechon, P. (2016). *Planification multirobot pour des missions de surveillance avec contraintes de communication*. PhD Thesis, Toulouse, ISAE.

[Beeson et al., 2005] Beeson, P., Jong, N. K., and Kuipers, B. (2005). Towards autonomous topological place detection using the extended voronoi graph. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4373–4379. IEEE.

[Beeson et al., 2010] Beeson, P., Modayil, J., and Kuipers, B. (2010). Factoring the Mapping Problem : Mobile Robot Map-building in the Hybrid Spatial Semantic Hierarchy. *The International Journal of Robotics Research*, 29(4) :428–459.

[Blochliger et al., 2018] Blochliger, F., Fehr, M., Dymczyk, M., Schneider, T., and Siegwart, R. (2018). Topomap : Topological Mapping and Navigation Based on Visual SLAM Maps. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*.

[Blum, 1967] Blum, H. (1967). A Transformation for Extracting New Descriptors of Shape. In Wathen-Dunn, W., editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge.

[Bonanni et al., 2014] Bonanni, T. M., Grisetti, G., and Iocchi, L. (2014). Merging partially consistent maps. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 352–363. Springer.

[Bu et al., 2016] Bu, S., Zhao, Y., Wan, G., and Liu, Z. (2016). Map2dfusion : Real-time incremental UAV image mosaicing based on monocular SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4564–4571. IEEE.

[Cadena et al., 2016] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, Present, and Future of Simultaneous Localization and Mapping : Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6) :1309–1332.

[Chaimowicz and Kumar, 2007] Chaimowicz, L. and Kumar, V. (2007). Aerial shepherds : Coordination among uavs and swarms of robots. In *Distributed Autonomous Robotic Systems 6*, pages 243–252. Springer.

[Chapoulie et al., 2012] Chapoulie, A., Rives, P., and Filliat, D. (2012). Topological segmentation of indoors/outdoors sequences of spherical views. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4288–4295. IEEE.

[Choset and Nagatani, 2001] Choset, H. and Nagatani, K. (2001). Topological simultaneous localization and mapping (SLAM) : toward exact localization without explicit localization. *IEEE Transactions on robotics and automation*, 17(2) :125–137.

[Churchill and Newman, 2013] Churchill, W. and Newman, P. (2013). Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14) :1645–1661.

[Drouilly, 2015] Drouilly, R. (2015). *Cartographie hybride metrique topologique et semantique pour la navigation dans de grands environnements*. PhD thesis, Nice.

[Drouilly et al., 2015] Drouilly, R., Rives, P., and Morisset, B. (2015). Hybrid metric-topological-semantic mapping in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5109–5114. IEEE.

[Dube et al., 2016] Dube, R., Gawel, A., Cadena, C., Siegwart, R., Freda, L., and Gianni, M. (2016). 3d localization, mapping and path planning for search and rescue operations. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 272–273. IEEE.

[Elfes et al., 1999] Elfes, A., Bergerman, M., Carvalho, J. R. H., de Paiva, E. C., Ramaos, J. J. G., and Bueno, S. S. (1999). Air-ground robotic ensembles for cooperative applications : Concepts and preliminary results. In *2nd International Conference on Field and service Robotics, Pittsburgh, Pa (USA)*, pages 75–80.

[Fankhauser et al., 2016] Fankhauser, P., Bloesch, M., Krusi, P., Diethelm, R., Wermelinger, M., Schneider, T., Dymczyk, M., Hutter, M., and Siegwart, R. (2016). Collaborative navigation for flying and walking robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2859–2866. IEEE.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

[Farazmanesh and Tavakoli, 2018] Farazmanesh, D. and Tavakoli, A. (2018). Morphological Decomposition and Compression of Binary Images via a Minimum Set Cover Algorithm. *Journal of Mathematical Imaging and Vision*.

[Ferrao et al., 2017] Ferrao, V. T., Vinhal, C. D. N., and da Cruz, G. (2017). An Occupancy Grid Map Merging Algorithm Invariant to Scale, Rotation and Translation. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 246–251, Uberlandia. IEEE.

[Forster et al., 2013] Forster, C., Pizzoli, M., and Scaramuzza, D. (2013). Air-ground localization and map augmentation using monocular dense reconstruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3971–3978. IEEE.

[Frietsch et al., 2008] Frietsch, N., Meister, O., Schlaile, C., and Trommer, G. F. (2008). Teaming of an ugv with a vtol-uav in urban environments. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1278–1285. IEEE.

[Garzon et al., 2013] Garzon, M., Valente, J., Zapata, D., and Barrientos, A. (2013). An Aerial-Ground Robotic System for Navigation and Obstacle Mapping in Large Outdoor Areas. *Sensors (Basel)*, 13(1) :1247–1267.

[Garzon Oviedo et al., 2012] Garzon Oviedo, M., Pereira Valente, J. R., Zapata, D., Chill, R., and Barrientos Cruz, A. (2012). Towards a ground navigation system based in visual feedback provided by a mini UAV. In *IEEE Intelligent Vehicles Symposium Workshops*.

[Gerkey and Mataric, 2002] Gerkey, B. and Mataric, M. (2002). Sold ! : auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5) :758–768.

[Graphm, 2009] Graphm (2009). GraphM package : approximate graph matching algorithms. Technical report.

[Grisetti et al., 2007] Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1) :34–46.

[Guastella et al., 2017] Guastella, D. C., Cantelli, L., Melita, C. D., and Muscato, G. (2017). A Global Path Planning Strategy for a UGV from Aerial Elevation Maps for Disaster Response :. SCITEPRESS - Science and Technology Publications.

[Guerin et al., 2015] Guerin, F., Guinand, F., Brethe, J.-F., Pelvillain, H., and others (2015). UAV-UGV cooperation for objects transportation in an industrial area. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 547–552. IEEE.

[Hackeloeer et al., 2013] Hackeloeer, A., Klasing, K., Krisp, J. M., and Meng, L. (2013). Comparison of Point Matching Techniques for Road Network Matching. *International archives of the photogrammetry remote sensing and spatial information sciences, XL-2W1*, pages 87–92.

Licence CC BY-NC-ND 3.0

[Harik et al., 2015] Harik, E. H. C., Guinand, F., Pelvillain, H., Guerin, F., and Brethe, J.-F. (2015). A decentralized interactive architecture for aerial and ground mobile robots cooperation. In *ICCAR*, pages 37–43. IEEE.

[Heffley, 1972] Heffley, D. R. (1972). The Quadratic Assignment Problem : A Note. *Econometrica*, 40(6) :1155.

[Ho and Newman, 2006] Ho, K. L. and Newman, P. (2006). Loop closure detection in SLAM by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9) :740–749.

[Huang and Beevers, 2005] Huang, W. H. and Beevers, K. R. (2005). Topological Map Merging. *The International Journal of Robotics Research*, 24(8) :601–613.

[IEEE, 2015] IEEE (2015). *IEEE Standard for Robot Map Data Representation for Navigation.*

[J. Delmerico and D. Scaramuzza, 2017] J. Delmerico and D. Scaramuzza (2017). Active Autonomous Aerial Exploration for Ground Robot Path Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

[James D, 2005] James D, M. (2005). *Stupid robot tricks : a behavior-based distributed algorithm library for programming swarms of robots.* PhD thesis.

[Kaess et al., 2008] Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM : Incremental Smoothing and Mapping. *IEEE Transactions on Robotics*, 24(6) :1365–1378.

[Kegl and Krzyzak, 2002] Kegl, B. and Krzyzak, A. (2002). Piecewise linear skeletonization using principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :59–74.

[Kloder and Hutchinson, 2006] Kloder, S. and Hutchinson, S. (2006). Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics*, 22(4) :650–665.

[Korrapati and Mezouar, 2014] Korrapati, H. and Mezouar, Y. (2014). Vision-based sparse topological mapping. *Robotics and Autonomous Systems*, 62(9) :1259–1270.

[Kuhn, 1955] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 :83–97.

[Kuipers, 1978] Kuipers, B. J. (1978). Modeling Spatial Knowledge. Technical report.

[Kuipers, 1991] Kuipers, B. J. (1991). A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Journal of Robotics and Autonomous Systems.*

[Kuipers, 2004] Kuipers, B. J. (2004). Using the Topological Skeleton for Scalable Global Metrical Map-Building. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Licence CC BY-NC-ND 3.0

[L. Cantelli, 2013] L. Cantelli (2013). UAV/UGV cooperation for surveying operations in humanitarian demining. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE.

[Labbani-Igbida et al., 2011] Labbani-Igbida, O., Charron, C., and Mouaddib, E. M. (2011). Haar invariant signatures and spatial recognition using omnidirectional visual information only. *Autonomous Robots*, 30(3) :333–349.

[Lau et al., 2013] Lau, B., Sprunk, C., and Burgard, W. (2013). Efficient grid-based spatial representations for robot navigation in dynamic environments. *Robotics and Autonomous Systems*, 61(10) :1116–1130.

[Leonard and Durrant-Whyte, 1991] Leonard and Durrant-Whyte (1991). Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems. Intelligence for Mechanical Systems, (IROS)*, pages 1442–1447. IEEE.

[Leordeanu and Hebert, 2005] Leordeanu, M. and Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1482–1489. IEEE.

[Li et al., 2016] Li, J., Deng, G., Luo, C., Lin, Q., Yan, Q., and Ming, Z. (2016). A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems. *IEEE Transactions on Vehicular Technology*, 65(12) :9585–9596.

[Lisien et al., 2005] Lisien, B., Morales, D., and Silver, D. (2005). The hierarchical atlas. *IEEE Transactions on Robotics*, 21(3) :473 – 481.

[Lowry et al., 2016] Lowry, S., Sunderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., and Milford, M. J. (2016). Visual Place Recognition : A Survey. *IEEE Transactions on Robotics*, 32(1) :1–19.

[Lu and Milios, 1997] Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4) :333–349.

[Luan, 2012] Luan, X. C. (2012). A structure-based approach for matching road junctions with different coordinate systems. In *Proceedings of the Twenty-second ISPRS Congress*.

[Mahendran et al., 2013] Mahendran, A., Dewan, A., Soni, N., and Krishna, K. M. (2013). UGV-MAV Collaboration for Augmented 2d Maps. pages 1–6. ACM Press.

[Marie, 2014] Marie, R. (2014). *Exploration autonome et construction de cartes topologiques referencees vision omnidirectionnelle*. PhD thesis.

[Marie et al., 2013] Marie, R., Labbani-Igbida, O., and Mouaddib, E. M. (2013). The Delta-Medial Axis : A robust and linear time algorithm for Euclidian skeleton computation. In *IEEE International Conference on Image Processing (ICIP)*, pages 3523–3526. IEEE.

[Marie et al., 2018] Marie, R., Said, H. B., Stephant, J., and Labbani-Igbida, O. (2018). Visual servoing on the generalized voronoi diagram using an omnidirectional camera. *Journal of Intelligent & Robotic Systems*.

Licence CC BY-NC-ND 3.0

[Mathews et al., 2012] Mathews, N., Christensen, A. L., O'Grady, R., and Dorigo, M. (2012). Spatially targeted communication and self-assembly. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2679. IEEE.

[Merveilleux et al., 2011] Merveilleux, P., Labbani-Igbida, O., and Mouaddib, E. M. (2011). Robust free space segmentation using active contours and monocular omnidirectional vision. In *IEEE International Conference on Image Processing (ICIP)*, pages 2877–2880. IEEE.

[Milford and Wyeth, 2012] Milford, M. J. and Wyeth, G. F. (2012). SeqSLAM : Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1643–1649. IEEE.

[Montek Singh et al., 1997] Montek Singh, Chatterjee, Amitabha, and Chaudhury, Santanu (1997). Matching structural shape descriptions using genetic algorithms. *Pattern Recognition*.

[Mourikis and Roumeliotis, 2007] Mourikis, A. I. and Roumeliotis, S. I. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE international conference on Robotics and automation, (ICRA)*, pages 3565–3572. IEEE.

[Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM : A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5) :1147–1163.

[Oleynikova et al., 2017] Oleynikova, H., Taylor, Z., Fehr, M., Nieto, J., and Siegwart, R. (2017). Voxblox : Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[Parker, 2003] Parker (2003). *Multi-Robot Systems : From Swarms to Intelligent Automata*, volume II.

[Parker and Fang Tang, 2006] Parker, L. and Fang Tang (2006). Building Multirobot Coalitions Through Automated Task Solution Synthesis. *Proceedings of the IEEE*, 94(7) :1289–1305.

[Parker, 2008] Parker, L. E. (2008). Distributed intelligence : Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1) :5–14.

[Pfaltz and Rosenfeld, 1967] Pfaltz, J. L. and Rosenfeld, A. (1967). Computer representation of planar regions by their skeletons. *Communications of the ACM*, 10(2) :119–122.

[Pfingsthorn and Birk, 2014] Pfingsthorn, M. and Birk, A. (2014). Representing and solving local and global ambiguities as multimodal and hyperedge constraints in a generalized graph SLAM framework. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4276–4283. IEEE.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0

[Ratter and Sammut, 2015] Ratter, A. and Sammut, C. (2015). Local map based graph slam with hierarchical loop closure and optimisation. In *Australasian Conference on Robotics and Automation 2015*.

[Renaudeau et al., 2018] Renaudeau, B., Labbani-Igbida, O., and Mourioux, G. (2018). Hybrid map mosaicing : A novel approach for large area mapping. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 23–28. IEEE.

[Riccio et al., 2015] Riccio, F., Lazaro, M. T., Gemignani, G., and Nardi, D. (2015). Multi Robot Perception and Action : World Modeling and Task Allocation. In *RSS Workshop on Principle of multirobot systems*.

[Roldan et al., 2016] Roldan, J., Garcia-Aunon, P., Garzon, M., de Leon, J., del Cerro, J., and Barrientos, A. (2016). Heterogeneous Multi-Robot System for Mapping Environmental Variables of Greenhouses. *Sensors*, 16(7) :1018.

[Saska et al., 2012] Saska, M., Krajnik, T., and Pfeucil, L. (2012). Cooperative micro UAV-UGV autonomous indoor surveillance. In *International Multi-Conference on Systems, Signals and Devices (SSD)*, pages 1–6. IEEE.

[Saska et al., 2014] Saska, M., Krajnik, T., Vonasek, V., Kasl, Z., Spurny, V., and Preucil, L. (2014). Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups. *Journal of Intelligent & Robotic Systems*, 73(1-4) :603–622.

[Schmuck et al., 2016] Schmuck, P., Scherer, S. A., and Zell, A. (2016). Hybrid metric-topological 3d occupancy grid maps for large-scale mapping. *IFAC-PapersOnLine*, 49(15) :230–235.

[Sebastian and Kimia, 2001] Sebastian and Kimia (2001). Curves vs skeletons in object recognition. In *IEEE International Conference on Image Processing (ICIP)*. IEEE.

[Sebastien Bosch, 2007] Sebastien Bosch (2007). *Contribution a la modelisation d'environnements par vision monoculaire dans un contexte de robotique aero-terrestre*. PhD thesis.

[Smith and Cheeseman, 1986] Smith and Cheeseman (1986). On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*.

[Stephen Tully et al., 2012] Stephen Tully, George Kantor, and Howie Choset (2012). A unified Bayesian framework for global localization and SLAM in hybrid metric/topological maps. *IJRR*.

[Stroupe et al., 2006] Stroupe, A., Okon, A., Robinson, M., Huntsberger, T., Aghazarian, H., and Baumgartner, E. (2006). Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. *Autonomous Robots*, 20(2) :113–123.

[Tao et al., 2011] Tao, T., Tully, S., Kantor, G., and Choset, H. (2011). Incremental construction of the saturated-GVG for multi-hypothesis topological SLAM. In

Licence CC BY-NC-ND 3.0

IEEE International Conference on Robotics and Automation (ICRA), pages 3072–3077. IEEE.

[Thrun and Bucken, 1996] Thrun, S. and Bucken, A. (1996). Learning Maps for Indoor Mobile Robot Navigation. Technical report, Carnegie-mellon univ Pittsburgh pa dept of computer science.

[Tipaldi and Arras, 2010] Tipaldi, G. D. and Arras, K. O. (2010). Flirt-interest regions for 2d range data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3616–3622. IEEE.

[Tuci et al., 2006] Tuci, E., Gro\s s, R., Trianni, V., Mondada, F., Bonani, M., and Dorigo, M. (2006). Cooperation through self-assembly in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(2) :115–150.

[Veloso and Stone, 1998] Veloso, M. and Stone, P. (1998). Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. In *IEEE International Conference on Multi Agent Systems*, pages 309–316. IEEE.

[Vidal et al., 2011] Vidal, T., Berger, C., Sola, J., and Lacroix, S. (2011). Environment modeling for cooperative aerial/ground robotic systems. In *Robotics research*, pages 681–696. Springer.

[Wang et al., 2007] Wang, C.-C., Thorpe, C., Thrun, S., Hebert, M., and Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9) :889–916.

[Zaslavskiy et al., 2009] Zaslavskiy, M., Bach, F., and Vert, J.-P. (2009). A Path Following Algorithm for the Graph Matching Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12) :2227–2242.

[Zlot and Stentz, 2006] Zlot, R. and Stentz, A. (2006). Market-based Multirobot Coordination for Complex Tasks. *The International Journal of Robotics Research*, 25(1) :73–101.

Brice Renaudeau — Thèse de doctorat — Université de Limoges
Licence CC BY-NC-ND 3.0