Direct visual odometry and dense large-scale environment mapping from panoramic RGB-D images

Renato Martins

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

Direct visual odometry and dense large-scale environment mapping from panoramic RGB-D images

Odométrie visuelle directe et cartographie dense de grands environnements à base d'images panoramiques RGB-D

École doctorale n°84

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Spécialité CONTRÔLE, OPTIMISATION, PROSPECTIVE

COMPOSITION DU JURY :

M. Cédric DEMONCEAUX
Univ. Bourgogne Franche-Comté, Rapporteur

M. Josechu GUERRERO
Universidad de Zaragoza, Rapporteur

M. Philippe MARTINET
Ecole Centrale de Nantes, Président du jury

M. Alessandro CORREA VICTORINO
Univ. Federal de Minas Gerais, Examinateur

M. Florent LAFARGE
Inria TITANE, Examinateur

M. El Mustapha MOUADDIB
Univ. de Picardie Jules Verne, Examinateur

M. Patrick RIVES
Inria LAGADIC, Examinateur

Soutenue par **Renato MARTINS**
le 27 Octobre 2017

Dirigée par **Patrick RIVES**

MINES ParisTech | PSL RESEARCH UNIVERSITY PARIS

Direct visual odometry and dense large-scale environment mapping from panoramic RGB-D images

Renato José Martins

Inria Sophia Antipolis — MINES ParisTech

September, 2017

Abstract

This thesis is in the context of self-localization and 3D mapping from RGB-D cameras for mobile robots and autonomous systems. We present image alignment and mapping techniques to perform the camera localization (tracking) notably for large camera motions or low frame rate. Possible domains of application are virtual and augmented reality, localization of autonomous vehicles or in 3D reconstruction of environments. We propose a consistent localization and 3D dense mapping framework considering as input a sequence of RGB-D images acquired from a mobile platform. The core of this framework explores and extends the domain of applicability of direct/dense appearance-based image registration methods. With regard to feature-based techniques, direct/dense image registration (or image alignment) techniques are more accurate and allow us a more consistent dense representation of the scene. However, these techniques have a smaller domain of convergence and rely on the assumption that the camera motion is small.

In the first part of the thesis, we propose two formulations to relax this assumption. Firstly, we describe a fast pose estimation strategy to compute a rough estimate of large motions, based on the normal vectors of the scene surfaces and on the geometric properties between the RGB-D images. This rough estimation can be used as initialization to direct registration methods for refinement. Secondly, we propose a direct RGB-D camera tracking method that exploits adaptively the photometric and geometric error properties to improve the convergence of the image alignment.

In the second part of the thesis, we propose techniques of regularization and fusion to create compact and accurate representations of large scale environments. The regularization is performed from a segmentation of spherical frames in piecewise patches using simultaneously the photometric and geometric information to improve the accuracy and the consistency of the scene 3D reconstruction. This segmentation is also adapted to tackle the non-uniform resolution of panoramic images. Finally, the regularized frames are combined to build a compact keyframe-based map composed of spherical RGB-D panoramas optimally distributed in the environment. These representations are helpful for autonomous navigation and guiding tasks as they allow us an access in constant time with a limited storage which does not depend on the size of the environment.

Keywords: RGB-D registration; mapping; visual odometry; visual SLAM; panoramic images

Résumé

Cette thèse se situe dans le domaine de l'auto-localisation et de la cartographie 3D pour des robots mobiles et des systèmes autonomes avec des caméras RGB-D. Nous présentons des techniques d'alignement d'images et de cartographie pour effectuer la localisation d'une caméra (suivi), notamment pour des caméras avec mouvements rapides ou avec faible cadence. Les domaines d'application possibles sont la réalité virtuelle et augmentée, la localisation de véhicules autonomes ou la reconstruction 3D des environnements. Nous proposons un cadre consistant et complet au problème de localisation et cartographie 3D à partir de séquences d'images RGB-D acquises par une plateforme mobile. Ce travail explore et étend le domaine d'applicabilité des approches de suivi direct dites "appearance-based". Vis-à-vis des méthodes fondées sur l'extraction de primitives, les approches directes permettent une représentation dense et plus précise de la scène mais souffrent d'un domaine de convergence plus faible nécessitant une hypothèse de petits déplacements entre images.

Dans la première partie de la thèse, deux contributions sont proposées pour augmenter ce domaine de convergence. Tout d'abord une méthode d'estimation des grands déplacements est développée s'appuyant sur les propriétés géométriques des cartes de profondeurs contenues dans l'image RGB-D. Cette estimation grossière (rough estimation) peut être utilisée pour initialiser la fonction de coût minimisée dans l'approche directe. Une seconde contribution porte sur l'étude des domaines de convergence de la partie photométrique et de la partie géométrique de cette fonction de coût. Il en résulte une nouvelle fonction de coût exploitant de manière adaptative l'erreur photométrique et géométrique en se fondant sur leurs propriétés de convergence respectives.

Dans la deuxième partie de la thèse, nous proposons des techniques de régularisation et de fusion pour créer des représentations précises et compactes de grands environnements. La régularisation s'appuie sur une segmentation de l'image sphérique RGB-D en patchs utilisant simultanément les informations géométriques et photométriques afin d'améliorer la précision et la stabilité de la représentation 3D de la scène. Cette segmentation est également adaptée pour la résolution non uniforme des images panoramiques. Enfin les images régularisées sont fusionnées pour créer une représentation compacte de la scène, composée de panoramas RGB-D sphériques distribués de façon optimale dans l'environnement. Ces représentations sont particulièrement adaptées aux applications de mobilité, tâches de navigation autonome et de guidage, car elles permettent un accès en temps constant avec une faible occupation de mémoire qui ne dépendent pas de la taille de l'environnement.

Mots clés: Recalage d'images; cartographie; odométrie visuelle; localisation; SLAM visuel; images panoramiques

Acknowledgments

Firstly, I would like to thank Cédric Demonceaux and Josechu Guerrero for examining my thesis work and Philippe Martinet, Alessandro Correa Victorino, Florent Lafarge and El Mustapha Mouaddib to be part of my thesis committee. In special, I would like to thank my supervisor, Patrick Rives, for sharing his ideas, advises and for his constant motivation. More than scientific directions to my work, his vision, expertise and motivation were and will remain an example and source of inspiration to my future career. Merci Patrick!

Being at Inria allowed me to have the privilege of meeting and collaborate with very talented researchers. Thanks to Eduardo Fernandez-Moral, Paolo Salaris and Tawsif Gokhool for the collaborations, discussions and the proof reading of my papers. But also for the other members of the Lagadic team in Sophia Antipolis and in Rennes, notably Alejandro Perez-Yus, Noel Meriaux, Panagiotis Papadakis, Dayana Hassan, Romain Drouilly and Denis Wolf. I could not forget the discussions and friendship with the Hephaistos team members, in special Artem Melnyk, Alain Coulbois and Mohamed Hedi Amri; and the support from my previous and current collaborators at Unicamp and at CTI Renato Archer, in special Samuel Siqueira Bueno for the encouragement and friendship during these last years.

I would like to thank Valérie Roy and the Ecole des Mines members in Sophia Antipolis, for the excellent professional training and for their administrative support. I gratefully acknowledge the financial support from CNPq (216026/2013-0), Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos - INCT-SEC (CNPq 573963/2008-8 e FAPESP 08/57870-9) and from Inria.

Last but not least, a special thanks to my family, my parents, brothers, Vanessa and Frédéric, por tudo que vocês fizeram e fazem por mim.

"Ce n'est pas une image juste, c'est juste une image"
Le Vent d'est (Jean-Luc Godard)

Contents

List of Figures

Chapter 5

List of Tables

List of Acronyms

DOF	Degrees of freedom
FOV	Field of view
ICP	Iterative closest point
MAD	Median of absolute differences
NCC	Normalized cross correlation
PDF	Probability density function
ROF	Rudin-Osher-Fatemi energy model
ROI	Region of interest
SfM	Structure from Motion
SLAM	Simultaneous localization and mapping
SNR	Signal-to-noise ratio
SSD	Sum of squared differences
SVD	Singular value decomposition
TSDF	Truncated signed distance functions
TV	Total variation
UAV	Unmaned autonomous vehicle
VSLAM	Visual simultaneous localization and mapping

Introduction Générale

1 Contexte et Motivation

La plupart des applications en robotique autonome nécessite de résoudre des problèmes de perception difficiles. Dans le contexte des robots mobiles autonomes, les problèmes de perception se décline en deux tâches principales : la localisation du robot et la cartographie de l'environnement. Afin d'accomplir ces tâches, les systèmes robotiques peuvent exploiter une grande variété de capteurs pour percevoir l'environnement et l'état du robot tels les capteurs extéroceptifs (comme les caméras, LIDAR, sonars), proprioceptifs (comme les dispositifs de mesure inertielle, ou encodeurs) ou des capteurs absolus (par example le GPS). Les applications potentielles couvrent de nombreux domaines : les véhicules intelligents et autonomes, l'agriculture, la sécurité, la réalité augmentée, l'architecture et la surveillance des environnements. Au cours des dernières décennies, les capteurs de vision ont été largement utilisés pour effectuer ces tâches car ils présentent de nombreux avantages. Ce sont des capteurs compacts, relativement peu coûteux qui peuvent fournir de nombreuses informations denses sur l'environnement telles que la couleur, la texture et la structure de la scène 3D. Des résultats récents impressionnants ont été obtenus, par exemple, dans les cas de l'exploration de Mars en utilisant deux caméras pour l'odométrie visuelle ou de la navigation autonome des véhicules (sur Terre) à l'aide de la base de données d'images Street View de Google. Plus récemment, l'apparition de caméras RGB-D (fournissant des informations sur la couleur (RGB) et la profondeur (D)) ouvre de nouvelles perspectives à ces applications. Malgré ces avancées, percevoir la structure de l'environnement et les mouvements des objets à partir des images, reste un domaine de recherche actif, en partie parce que seule une observation échantillonnée 2D/2D + t de la scène est accessible via le processus de formation de l'image. Dans le cas d'applications de robotique mobile, cette information n'est pas suffisante et une estimation précise de la localisation de la caméra et une reconstruction précise des modèles photométriques et géométriques de la scène, s'avèrent nécessaires. Alors que la plupart des méthodes utilisées pour résoudre ces problèmes sont basées sur la sélection et la mise en correspondance d'éléments caractéristiques de l'image (techniques basées *features*), des méthodes plus récentes (appelées méthodes directes ou *appearance-based*) utilisent tout le contenu de l'image sans effectuer explicitement l'extraction de caractéristiques ou la correspondance. L'enregistrement direct et les méthodes de cartographie dense ont connu

une utilisation croissante au cours des dernières années et se sont révélés plus précis par rapport aux techniques basées sur les features. Ces méthodes, cependant, ont leurs limites, notamment au niveau de leur faible domaine de convergence qui limite souvent leur application.

2 Objetifs

En considérant des environnements réels complexes, les tâches de cartographie et de suivi des caméras doivent explicitement prendre en compte les grands mouvements de la caméra, des scènes à grande échelle et la variabilité des conditions d'apparence dans un modèle photométrique et géométrique de l'environnement. Les objectifs de cette thèse sont donc centrés autour de la conception de méthodes d'enregistrement direct efficaces et robustes associées à des représentations cartographiques adaptées aux applications visées. Nous explorons le potentiel de caméras avec un large champ de vision afin d'augmenter le bassin de convergence et de construire des représentations denses et précises d'environnement à grande échelle. Nous proposons de modéliser l'environnement en utilisant localement des images sphériques egocentrées, positionnées dans une structure topologique de graphe couvrant de façon optimale la scène. Cette représentation possède des propriétés intéressantes vis à vis de nos applications. Tout d'abord, l'apparence visuelle de la scène est invariante aux rotations dans le cas d'images sphériques panoramiques. Deuxièmement, l'imagerie sphérique généralise d'autres images avec large champ de vision telles que celles acquises par des caméras catadioptriques omnidirectionnelles ou fisheye. Un autre champ d'investigation de cette thèse est d'étudier comment représenter efficacement ce modèle, c'est à dire de façon compacte et précise. Ce sont des aspects clés pour générer des modèles capables d'être utilisés plus tard dans la navigation autonome du robot ou dans le rendu virtuel des scènes.

3 Contributions

Dans cette thèse, nous avons choisi de traiter un aspect particulier qui limite drastiquement l'applicabilité des méthodes d'enregistrement direct d'images, à savoir la faiblesse du domaine de convergence. L'élargissement de ce domaine présente un intérêt pratique non seulement pour les capteurs RGB-D sphériques, mais aussi pour toutes les approches d'odométrie visuelle basées sur l'enregistrement direct. Les contributions portent sur deux aspects :

— **L'augmentation de la robustesse du suivi en cas des grands déplacements de la caméra**. Les algorithmes d'enregistrement direct couramment utilisés ont une convergence locale et reposent sur l'hypothèse que le mouvement entre les images est petit ou qu'il existe une bonne estimation initiale de la pose à partir d'un capteur externe (par exemple, de l'odométrie des roues ou des dispositifs de mesure inertielle). Nous proposons deux formulations pour assouplir cette hypothèse. Tout d'abord, nous avons développé

une stratégie d'estimation de pose efficace pour calculer une estimation approximative des mouvements importants entre les images de profondeur, qui peut être utilisé comme initialisation pour les méthodes d'enregistrement direct. La rotation et la translation sont calculées d'une manière séquentielle découplée. Deuxièmement, nous décrivons une technique de suivi de caméra RGB-D qui exploite de manière adaptative les images photométriques et géométriques en fonction de leurs caractéristiques de convergence. Ces contributions nous permettent d'effectuer le suivi d'une caméra soumise à de grandes rotations et translations ainsi que des occlusions dans de vrais scénarios d'intérieur et d'extérieur.

— **Représentation cartographique dense adaptée à la localisation avec vision**. En ce qui concerne la cartographie, nous proposons une régularisation qui explore simultanément l'information photométrique et géométrique de la scène pour améliorer la précision et l'apparence de la structure 3D des images. Ceci est particulièrement important lors de l'utilisation des images de profondeur provenant de la stéréo. La régularisation s'effectue à partir d'une segmentation des images en patchs planaires à l'aide de la couleurs et des normales. Cette segmentation est basée sur une technique superpixel SLIC et considère la résolution non uniforme des images panoramiques. La dernière contribution est une technique de cartographie compacte basée sur des images clés en utilisant un partitionnement de l'espace navigable de l'environnement. Les images améliorées sont ensuite combinées pour créer une carte d'image-clés compacte, tout en conservant des caractéristiques intéressantes pour la navigation et la localisation en utilisant l'enregistrement direct. Concrètement, nous avons observé que ces techniques ont ajouté plusieurs avantages pour le suivi des caméras et la compacité de la carte, en augmentant sa précision et sa consistence.

3.1 Publications

Cette thèse a conduit à cinq publications internationales dans des conférences de robotique et de traitement d'images :

— [Martins et al., 2017] R. Martins, E. Fernandez-Moral and P. Rives. **"An efficient rotation and translation decoupled initialization from large field of view depth images"**. IEEE International Conference on Intelligent Robots and Systems, IROS 2017.

— [Martins et al., 2016] R. Martins, E. Fernandez-Moral and P. Rives. **"Adaptive direct RGB-D registration and mapping for large motions"**. Asian Conference on Computer Vision, ACCV 2016.

— [Martins and Rives, 2016] R. Martins and P. Rives. **"Increasing the convergence**

domain of RGB-D direct registration methods for vision-based localization in large scale environments". IEEE Intelligent Transportation Systems Conference Workshop on Planning, Perception and Navigation for Intelligent Vehicles, ITSC PPNIV 2016.

— [Martins et al., 2015] R. Martins, E. Fernandez-Moral and P. Rives. **"Dense accurate urban mapping from spherical RGB-D images"**. IEEE International Conference on Intelligent Robots and Systems, IROS 2015.

— [Gokhool et al., 2015] T. Gokhool, R. Martins, P. Rives and N. Despre. **"A compact spherical RGBD keyframe-based representation"**. IEEE International Conference on Robotics and Automation, ICRA 2015.

J'ai également collaboré avec d'autres membres de l'équipe Lagadic Sophia Antipolis sur les mesures d'évaluation pour la segmentation sémantique, ce qui a abouti à une publication non incluse dans ce manuscrit :

— [Fernandez-Moral et al., 2017] E. Fernandez-Moral, R. Martins, D. Wolf and P. Rives. **"A new metric for evaluating semantic segmentation : leveraging global and contour accuracy"**. IEEE International Conference on Intelligent Robots and Systems Workshop on Planning, Perception and Navigation for Intelligent Vehicles, IROS PPNIV 2017.

En terme de développements logiciels, une bibliothèque avec les implémentations Matlab et C ++ des modules principaux [1] a été développée qui va être mise prochainement à disposition de la communauté de robotique.

4 Structure de la thèse

Le manuscrit se compose de six chapitres répartis en trois parties. La première partie présente les informations de base nécessaires à la compréhension de cette thèse, tels que les concepts fondamentaux d'imagerie et les travaux connexes à la localisation visuelle et à la cartographie. La deuxième partie présente des techniques pour augmenter la convergence des méthodes d'enregistrement direct RGB-D. Dans la dernière partie, nous formulons le problème de la construction d'une représentation ccompacte de la scène en utilisant les techniques d'enregistrement d'image améliorées susmentionnées. Une brève description de ces chapitres est la suivante :

1. Quelques exemples et modules sont accessibles sur https://github.com/omni-rgbd/

Partie I

Chapitre 2 : Vision Panoramique

Ce chapitre décrit les concepts fondamentaux d'imagerie utilisés le long de la thèse. Il contient la représentation et les capteurs utilisées pour construire les images panoramiques. Nous présentons également les caméras avec grand champ de vision et certaines de ses propriétés dans des contextes de suivi, de localisation et de cartographie.

Chapitre 3 : Enregistrement d'Images et Travaux Associés

Dans ce chapitre, nous donnons un aperçu des travaux connexes sur l'enregistrement des images et le contexte technique des méthodes directes. Les points forts et les limites de certaines techniques pertinentes sont également discutés.

Partie II

Chapitre 4 : Initialisation Efficace de Pose à partir de Vecteurs Normaux

Ce chapitre décrit une technique d'enregistrement utilisant les vecteurs normaux des images en profondeur. La technique est calculée de manière découplée séquentielle (rotation puis translation), non itérative et avec un large domaine de convergence et peut donc être utilisée comme initialisation pour les méthodes directes d'enregistrement. Nous analysons les limites et l'efficacité de cette formulation et montrons les résultats de l'enregistrement à l'aide de séquences simulées et réelles acquises avec des caméras sphériques et fisheye.

Chapitre 5 : Enregistrement RGB-D Adaptif pour des Grands Déplacements

Ce chapitre propose une approche permettant d'accroitre le bassin de convergence des méthodes directes d'enregistrement d'images. Nous proposons une approche d'enregistrement adaptative en exploitant l'observation selon laquelle les termes d'erreur d'intensité et de profondeur affichent différentes propriétés de convergence pour des mouvements de petite et de grande taille. L'amélioration du bassin de convergence est démontrée par des séquences simulées et réelles, en utilisant des caméras sphériques et de perspective.

Partie III

Chapitre 6 : Régularisation d'Image en Patchs Planaires

Ce chapitre présente une approche de régularisation pour filtrer les images de profondeur, en particulier celles provenant de la stéréo. Nous proposons une segmentation en superpixels en utilisant à la fois la couleur et l'orientation de la surface pour contrôler la régularisation. Cette segmentation considère la résolution non uniforme des images panoramiques.

Chapitre 7 : Cartographie RGB-D Compacte pour une Visibilité Optimale de l'Environnement

Le dernier chapitre, plus prospectif, décrit une stratégie de cartographie compacte dédiée à la localisation de robots et à la navigation autonome. Ce cadre de cartographie compacte est basé sur des images clés et explore les approches proposées d'enregistrement et de régularisation pour améliorer les images clés et pour créer des représentations topologiques-métriques compactes garantissant la visibilité et la couverture de la scène. Nous montrons et discutons quelques résultats de cartographie préliminaires obtenus en environnement d'intérieur.

Chapitre 8 : Conclusions et Perspectives

Enfin, nous concluons le manuscrit avec un résumé et des perspectives.

Chapter 1

General Introduction

1.1 Context and Motivation

At the base of most applications in robotics lies a difficult perception problem. The core perception problem for autonomous mobile robots comprises two characteristic tasks: robot localization and environment mapping. In order to accomplish these tasks, robotic systems can exploit a wide variety of sensors to perceive the environment and its own states such as exteroceptive (e.g., cameras, LIDARs, Sonars), proprioceptive (e.g., inertial measurement devices, encoders) or absolute sensors (e.g., GPS). In the last decades, vision has been extensively used to perform these tasks, with applications in many areas including mobility, agriculture, security, augmented reality, architecture and environment monitoring. Among the many reasons for their wide usage, cameras are relatively low-cost sensors, compacts and can furnish extensive dense information about the environment such as color, texture and the 3D scene structure. Impressive recent milestones of such applications are the exploration of Mars using two cameras for visual odometry or the autonomous vehicle navigation (on Earth) using Google's Street View image database. Moreover, the appearing of RGB-D cameras (providing color (RGB) and depth (D) information) open new perspectives to these applications. Still, perceiving the environment structure and the objects' motions from images remains an active research domain, partially because only a sampled and 2D/2.5D projected observation is gathered from the superposition of different spectral and spatial phenomena during image formation. Conversely to humans, who cope with this by performing sensorial fusion, semantic and contextual information, robotic systems often require an accurate estimate of the camera location (called registration or tracking) and of the photometric and geometric scene models.

While most of the employed methods to track the camera and to build these models are based on the selection and matching of distinctive characteristics/features (feature-based techniques), direct (appearance-based) methods use all the image content without performing any explicit feature extraction or matching. Direct registration and dense mapping methods have seen an increasing usage in recent years and proved to be more accurate when compared to feature-

based techniques. These methods, however, have a smaller basin of convergence which often limits their application.

1.2 Objectives

As we consider complex real environments, the mapping and camera tracking tasks require explicitly taking into account large camera motions, large-scale scenes and the variability of viewing conditions within a photometric and geometric model of the environment. The objectives of this thesis are then centered around the design of direct registration techniques and mapping representations that are adapted to deal in such conditions. We explore the potential of wide field of view cameras to increase the basin of convergence and to build adapted mapping strategies. We propose to model the environment using ego-centric spherical images, which have nice properties for the camera tracking and mapping. Firstly, the visibility of the scene is invariant to rotations in panoramic spherical images. Secondly, spherical imagery generalizes other wide field of view images such as the ones acquired by fisheye and omnidirectional catadioptric cameras. An adjacent objective of this thesis is to investigate how to efficiently represent this model. These are key aspects for generating accurate models to be used later on in robot autonomous navigation or in scene rendering.

1.3 Contributions

In this thesis, we have chosen to focus on a peculiar aspect that drastically limits the applicability of direct image registration methods, namely the weakness of the convergence domain. Enlarging this domain is of practical interest not only for spherical RGB-D sensors but, potentially, for all the visual odometry approaches based on direct registration. The contributions can be synthesized in two fronts:

— **Increasing the robustness to large motions**. Commonly used direct registration algorithms have local convergence and rely on the assumption that the motion between the images is small or that there is a good initial estimation of the pose from an external sensor (e.g., from wheel odometry or inertial measurement devices). We propose two formulations to relax this assumption. First, we developed a fast pose estimation strategy to compute a rough estimate of large motions between wide field of view depth images, which can be used as initialization to direct registration methods. The rotation and translation are computed in a decoupled sequential way. Second, we describe an RGB-D camera tracking technique that exploits adaptively the photometric and geometric images based on their convergence characteristics. These contributions allow us to perform the tracking of the camera subjected to large rotations and translations, occlusions and moving objects in real indoor and outdoor scenarios.

— **Adapted dense mapping representation to vision-based localization**. Concerning the mapping, we propose a regularization that explores simultaneously the photometric and geometric information of the scene to improve the accuracy and the appearance of the 3D structure of frames. This is particularly relevant when using the stereo depth images. The regularization is performed from a segmentation of the frames in piecewise patches using both color and normal images. This segmentation is based on a state-of-the-art superpixel technique and considers the non-uniform resolution of panoramic images. The last contribution is a compact keyframe-based mapping technique using a partitioning of the free space of the environment. The improved frames are then combined to build the sparse keyframe map, while maintaining interesting characteristics for navigation and localization using direct registration. Concretely, we observed that these techniques added several advantages for the camera tracking and the compactness of the map, increasing its accuracy and consistency.

1.3.1 Publications

This thesis led to five international publications in robotics and computer vision conferences:

— [Martins et al., 2017] R. Martins, E. Fernandez-Moral and P. Rives. **"An efficient rotation and translation decoupled initialization from large field of view depth images"**. IEEE International Conference on Intelligent Robots and Systems, IROS 2017.

— [Martins et al., 2016] R. Martins, E. Fernandez-Moral and P. Rives. **"Adaptive direct RGB-D registration and mapping for large motions"**. Asian Conference on Computer Vision, ACCV 2016.

— [Martins and Rives, 2016] R. Martins and P. Rives. **"Increasing the convergence domain of RGB-D direct registration methods for vision-based localization in large scale environments"**. IEEE Intelligent Transportation Systems Conference Workshop on Planning, Perception and Navigation for Intelligent Vehicles, ITSC PPNIV 2016.

— [Martins et al., 2015] R. Martins, E. Fernandez-Moral and P. Rives. **"Dense accurate urban mapping from spherical RGB-D images"**. IEEE International Conference on Intelligent Robots and Systems, IROS 2015.

— [Gokhool et al., 2015] T. Gokhool, R. Martins, P. Rives and N. Despre. **"A compact spherical RGBD keyframe-based representation"**. IEEE International Conference on Robotics and Automation, ICRA 2015.

I also collaborated with other research group members on evaluation metrics for semantic segmentation, which resulted in one publication not included in this manuscript:

— [Fernandez-Moral et al., 2017] E. Fernandez-Moral, R. Martins, D. Wolf and P. Rives. **"A new metric for evaluating semantic segmentation: leveraging global and contour accuracy"**. IEEE International Conference on Intelligent Robots and Systems Workshop on Planning, Perception and Navigation for Intelligent Vehicles, IROS PPNIV 2017.

Finally, we build a library with Matlab and C++ implementations of the main modules[1]. We plan to make the library and its components available to the robotics community.

1.4 Thesis Structure

The core of the manuscript is composed of six chapters which are distributed in three parts. The first part introduces the necessary background information for the understanding of this thesis, such as basic imaging concepts and related works to visual localization and mapping. The second part presents techniques to increase the convergence of direct RGB-D registration methods. In the last part, we formulate the problem of building a compact map model of the scene using the aforementioned improved image registration techniques. A brief description of these chapters is as follows:

Part I

Chapter 2: Panoramic Vision

This chapter describes the basic imaging concepts used along the thesis. It contains the frame representation and the sensor acquisition rigs used to build the panoramic images. We also introduce wide field of view cameras and some of its properties in 3D based tracking, localization and mapping contexts.

Chapter 3: Image Registration and Overview of Related Works

In this chapter, we give an overview of related works on image registration and the technical background of direct methods. The strengths and limitations of some relevant techniques are also discussed.

1. Same examples and modules are accessible at https://github.com/omni-rgbd/.

Part II

Chapter 4: Efficient Pose Initialization from Normal Vectors

This chapter describes a registration technique using the normal vectors of depth images. The technique is computed in a decoupled way (first rotation and then the translation), not iterative and with a large convergence domain and therefore can be used with an initialization framework to direct registration methods. We analyze the limitations and the observabilty of this formulation and show registration results using simulated and real sequences acquired with spherical and fisheye cameras.

Chapter 5: Adaptive Direct RGB-D Registration for Large Motions

This chapter continues in the line of increasing the basin of convergence of direct image registration methods. We propose an adaptive registration approach exploring the observation that the intensity and depth error terms display different convergence properties for small and large motions. The improvement of the basin of convergence is demonstrated with simulated and real sequences, using spherical and perspective benchmark sequences.

Part III

Chapter 6: Frame Regularization in Piecewise Planar Patches

This chapter presents a regularization approach to filter the depth images, in special the ones using stereo matching. We propose a superpixel segmentation using both color and surface orientation to drive the regularization of the frame in planar patches. This segmentation is based on a state-of-the-art superpixel technique and considers the non-uniform resolution of panoramic images.

Chapter 7: RGB-D Compact Mapping for Optimal Environment Visibility

The last chapter describes a compact mapping strategy dedicated for robot localization and autonomous navigation. This compact mapping framework is based on keyframes and explores the proposed registration and regularization approaches to improve the keyframes and to build useful sparse topological-metric representations with increased visibility. We show and discuss preliminary mapping results of an indoor environment.

Chapter 8: Conclusions and Perspectives

Finally, we conclude the manuscript with a summary and perspectives.

It is worth mentioning that at the beginning of each chapter in this thesis, we review related works specific to the topics covered therein.

Part I

Background and Introduction to Direct Image Registration

Chapter 2

Panoramic Vision

Contents

2.1 Introduction

Panoramic images are defined as those images whose field of view (FOV) comprises 360 ° in the horizontal plane. The example of excellence of panoramic images is the spherical view. Such images have some important advantages in computer vision, since problems as optical flow, ego-motion estimation and place recognition are better conditioned. Furthermore, spherical vision provides a decoupling between rotation and translation flows, which can be exploited for camera tracking and scene reconstruction. These advantages have already been exploited during the last decades in mobile robotics for scene modelling (e.g., [Micušık et al., 2003, Dayoub et al., 2011]), visual odometry (e.g., [Scaramuzza and Siegwart, 2008, Zhang et al., 2016]), SLAM (e.g., [Kim and Chung, 2003, Chapoulie et al., 2011]), place recognition (e.g., [Ulrich and Nourbakhsh, 2000, Jogan and Leonardis, 2000]) or visual-based navigation (e.g., [Gaspar et al., 2000, Meilland et al., 2015]).

Recently, a new market of action cameras and virtual reality has appeared with several fully integrated devices capturing spherical panoramas in real time, such as the Nikon KeyMission 360, the Giroptic 360cam or the Ricoh Theta S, at a relatively low cost. Thanks to their light weight, such devices could be embedded on mobile robots or serve for personal guidance applications. However, most available spherical and panoramic images are still built by warping images from catadioptric and fisheye cameras [Pérez-Yus et al., 2016]; or by mosaicing a set of smaller FOV images from a rig of perspective cameras. A typical example is Google's R7 camera rig composed of fifteen perspective cameras [Anguelov et al., 2010], as shown in fig. 2.1. One should note also that although spherical imagery offers advantages for ego-motion estimation, it also comes with intrinsic difficulties such as shape distortions, non-uniform resolution and because most of the available image processing tools are conceived to perspective images.

This chapter presents the basic panoramic imaging concepts used throughout the manuscript. In an attempt of giving a concise and comprehensive exposition, we start by presenting camera projection models in section 2.2. Section 2.3 describes acquisition strategies for producing RGB-D spherical images and the solutions developed previously by our group. A review of some elementary properties of spherical vision is given in section 2.4, followed by the normal computation in section 2.5. Finally, we conclude the chapter with a brief discussion and closing remarks in section 2.6.

2.2 Image Representation and Projections

An image frame $\boldsymbol{\mathcal{S}}$ can be composed of an RGB image $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{I}} \in [0\ 1]^{m \times n \times 3}$ representing the scene photometric model, a depth image $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{D}} \in \mathbb{R}^{m \times n}$ encoding the scene geometry or both photometric and depth images, i.e., $\mathcal{S} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}\}$ for RGB-D frames. The image formation $\boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{D}}$, from the photometric and geometric models of the scene, depend on

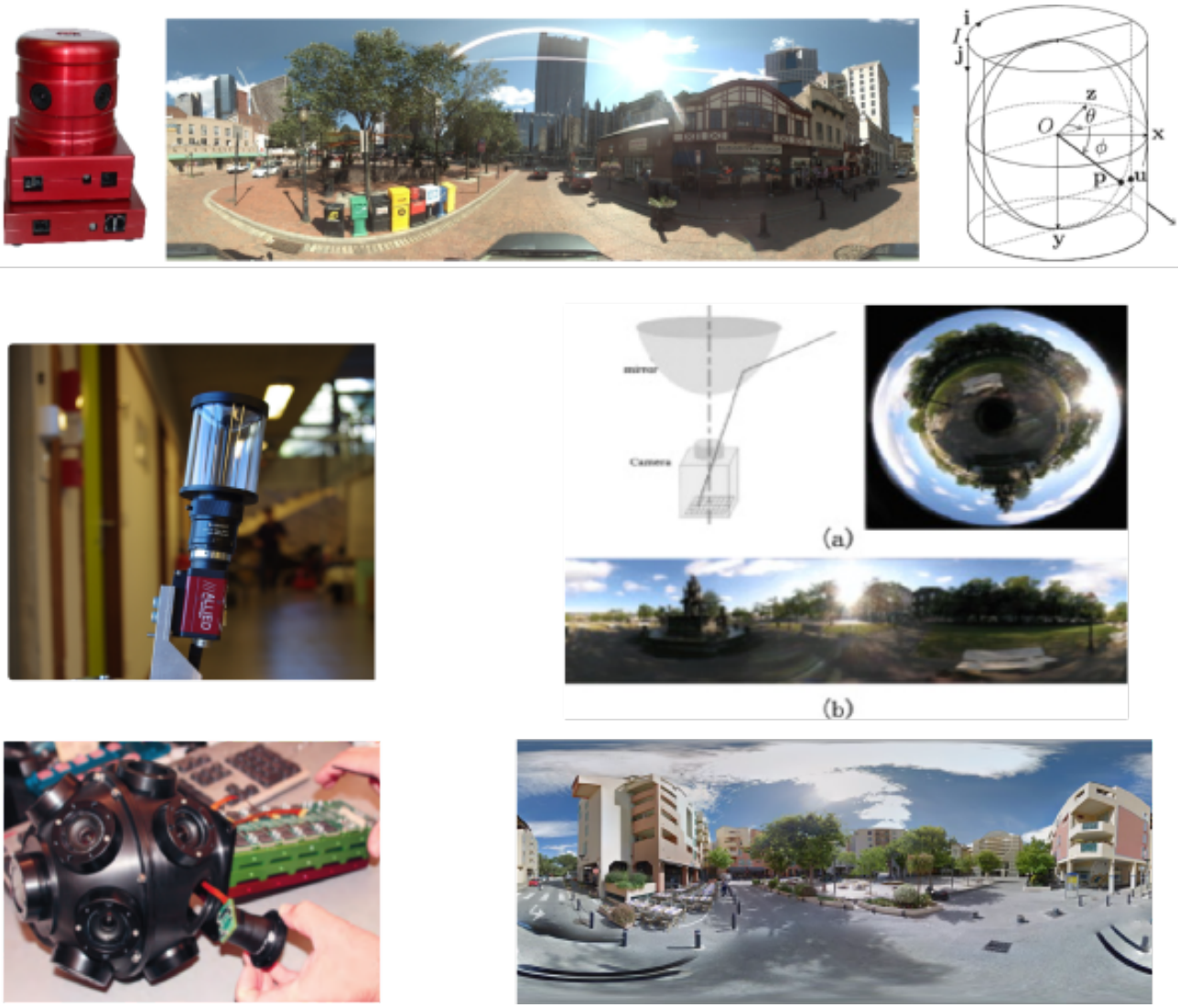

Figure 2.1 – First row: Point Grey Ladybug spherical camera and spherical panoramic image. Images courtesy of [Torii et al., 2009]. Second row: Omnidirectional catadioptric camera (left) and example of warped image in the sphere. Right images courtesy of [Liu et al., 2014]. Third row: Google's R7 camera rig with fifteen perspective cameras and Street View image from Garbejaire in Sophia Antipolis. Left image courtesy of [Anguelov et al., 2010].

many superposed complex phenomena, ranging from the physical and spectral quantities to the employed sensors. For instance, the color image $\boldsymbol{\mathcal{I}}$ depends on the surface texture, the illumination, the diffusion and the spectral reflections, the scene geometric model, the camera projection model and its point-of-view. We start by describing the projection models used in this thesis, other image formation topics are covered in more detail in computer vision books, e.g., [Hartley and Zisserman, 2003, Faugeras et al., 2001].

In order to recover the motion from images, we assume global shutter and central cameras [1]. Under this assumption, we can model wide field of view (FOV) images similarly to classic

1. Central cameras obeys the single viewpoint property, i.e., all the projection rays to form the image are constrained to meet at a single point.

perspective images thanks to a unified projection model [Geyer and Daniilidis, 2001, Barreto, 2006]. These wide FOV images (also known as ominidirectional) can be acquired using wide FOV sensors such as fisheye cameras, a combination of perspective cameras (e.g., [Anguelov et al., 2010, Meilland et al., 2011, Fernandez-Moral et al., 2014]) or using catadioptric devices, i.e., the acquisition sensor is composed of a mirror and a perspective/orthographic camera. All these images can be represented in the unit sphere under the assumption of central cameras through a calibration procedure. Several calibration techniques are available in the literature to obtain the calibration parameters (see [Mei and Rives, 2007] and [Puig et al., 2012] for a comparative study of different calibration algorithms). We present in fig. 2.1 some examples of wide FOV images and their respective acquisition systems. Hence, due to their importance to the following chapters, we recall the spherical and perspective projections.

2.2.1 Perspective Images and Pinhole Camera Model

Most commercial cameras can be described as pinhole cameras, which are modeled by a perspective projection. Pinhole cameras obeys the assumption of central cameras because all optical rays intersect in the camera optical center. Given a 3D point in the camera coordinate system $\mathbf{P} \in \mathbb{R}^3$ and using the perspective projection, the pixel coordinate in the image is given by:

$$\mathbf{p} = \mathbf{K}\|\mathbf{P}\|_P \text{ and } \|\mathbf{P}\|_P := \frac{\mathbf{P}}{\mathbf{e}_3^T\mathbf{P}} \in \mathbb{P}^2, \text{ for } \mathbf{e}_3^T\mathbf{P} \neq 0, \tag{2.1}$$

where $\| \ \|_P : \mathbb{R}^3 \rightarrow \mathbb{P}^2$ is the perspective normalization operator while $\mathbf{K}$ is the camera intrinsic parameters matrix such as:

$$\mathbf{K} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & fs & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

with (k_u, k_v) the scaling of pixel axes, f the focal length, (c_u, c_v) the coordinates of the image center in pixels and s is the parameter describing the skewness of the two image axes (in general $k_u \approx k_v$ and $s \approx 0$). $\mathbf{K}$ can be obtained using available camera calibration algorithms (e.g., [Zhang, 2000]). The perspective projection preserves straight lines and angles, i.e., straight lines in the scene are mapped to straight lines in the image. However, it stretches objects for FOV's wider than 90 degrees and it is not even defined for FOV's wider than 180 degrees – therefore the need of, at least, two planes to represent a panorama.

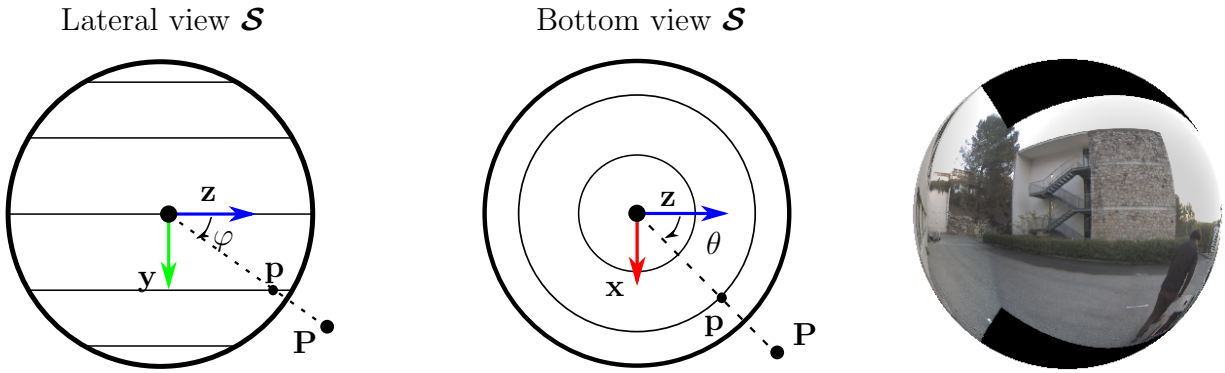

Figure 2.2 – Adopted coordinate system and spherical image. The first scheme shows the lateral view of the sphere and the second scheme depicts the bottom view. The respective positive spherical coordinates (φ, θ) for a given 3D point is depicted in each scheme.

2.2.2 Spherical Panoramic Images

Similarly to the perspective case, the pixel coordinates of the scene 3D point is obtained from the spherical projection model Π_S in the unit sphere:

$$\mathbf{p} = \Pi_S(\|\mathbf{P}\|_S), \tag{2.3}$$

where the spherical normalization operator $\| \ \|_S$: $\mathbb{R}^3_+ \rightarrow \mathbb{S}^2$ is defined as:

$$\|\mathbf{P}\|_S := \frac{\mathbf{P}}{\|\mathbf{P}\|_2} \in \mathbb{S}^2, \text{ for } \|\mathbf{P}\|_2 \neq 0. \tag{2.4}$$

This normalization will be extensively used in other contexts as, for instance, with normal vectors and rotations. The projection Π_S in (2.3) depends on the adopted spherical coordinate system and the sphere sub-sampling. The following spherical coordinates are adopted to parametrize the unit sphere:

$$\mathbb{S}^2 = \left\{ \Pi_S^{-1}(\mathbf{p}) := \begin{pmatrix} \sin(\theta(\mathbf{p}))\cos(\varphi(\mathbf{p})) \\ \sin(\varphi(\mathbf{p})) \\ \cos(\theta(\mathbf{p}))\cos(\varphi(\mathbf{p})) \end{pmatrix}, \theta(\mathbf{p}) \in [-\pi, \pi) \text{ and } \varphi(\mathbf{p}) \in [-\pi/2, \pi/2] \right\} \tag{2.5}$$

where the zenith/elevation (φ) and azimuth/longitude (θ) are the spherical coordinates for each pixel as shown in fig. 2.2.

2.2.2.1 Equirectangular Spherical Projection

The sphere is encoded in a 2D planar image using the equirectangular sub-sampling (as known as geographic projection), i.e., considering a constant solid angle between neighbouring pixels. Consequently, the relationship between the image pixel $\mathbf{p} \in \mathbb{P}^2$ to spherical coordinates $g_1 : \mathbf{p} = (u, v, 1) \rightarrow (\theta, \varphi, 1)$ is a bijective linear function that can be encoded by a constant

intrinsic matrix $\mathbf{K} \in \mathbb{R}^{(3\times3)}$. Therefore the spherical projection can be seen as a conversion from Cartesian $\|\mathbf{P}\|_S^T = (x\ y\ z)$ to spherical coordinates $(\theta, \varphi, 1)$ followed by a scaling into pixel coordinates $\mathbf{p}^T = (u, v, 1)$:

$$\mathbf{p} = \Pi_S(\|\mathbf{P}\|_S) = \mathbf{K} \begin{pmatrix} \arctan(x/z) \\ \arcsin(y/\sqrt{x^2+y^2+z^2}) \\ \sqrt{x^2+y^2+z^2} \end{pmatrix} \tag{2.6}$$

Observe that the uniform solid angle sampling generates a non-uniformly sampled sphere in the Euclidean 3D space, e.g., the regions near the poles are oversampled compared to regions neighbouring the equator. This projection (2.6) is also non-injective since any Cartesian point has infinitely many equivalent spherical coordinates $(\theta + 2\pi k_1, \varphi + 2\pi k_2, \bullet)$ for all $k_1, k_2 \in \mathbb{Z}$. Furthermore, the projection cannot be established for the poles $(\bullet, \pm\pi/2, \bullet)$ and the origin.

Other image representations can be used to represent wide FOV images besides the equirectangular and perspective, e.g., cubic, cylindrical, Mercator and stereographic. For instance, the stereographic projection of the spherical images will be used throughout this manuscript for visualization purposes. We show in figs. 2.3 and 2.4 typical examples of panoramic images. The reader can see [Zelnik-Manor et al., 2005, Sacht et al., 2010] for more details about these projections and their limitations/characteristics.

2.3 RGB-D Imagery

As introduced in section 2.1, RGB-D image frames encode both the photometry and 3D geometry of the scene. Therefore, an RGB-D image allows to perform realistic virtual view synthesis in a local domain around the point of view of the camera. This section discuss how to obtain the depth image $\boldsymbol{\mathcal{D}}$ related to each intensity image. We consider two solutions to estimate the depth: *i)* using active sensors such as, for instance, LIDARs and infrared cameras and; *ii)* stereo matching.

2.3.1 Depth from Active Sensors

In the case of using active sensors, only an extrinsic calibration between the camera and the active sensor is required. This calibration can be divided in two main groups: overlapping cameras (e.g., using [Zhang and Pless, 2004, Mei and Rives, 2006, Vasconcelos et al., 2012]) and non-overlapping cameras (e.g., [Lébraly et al., 2010, Fernandez-Moral et al., 2014]). Therefore, the central projection assumption and the pixel correspondence in the intensity and depth images are closely related to the accuracy of the calibration.

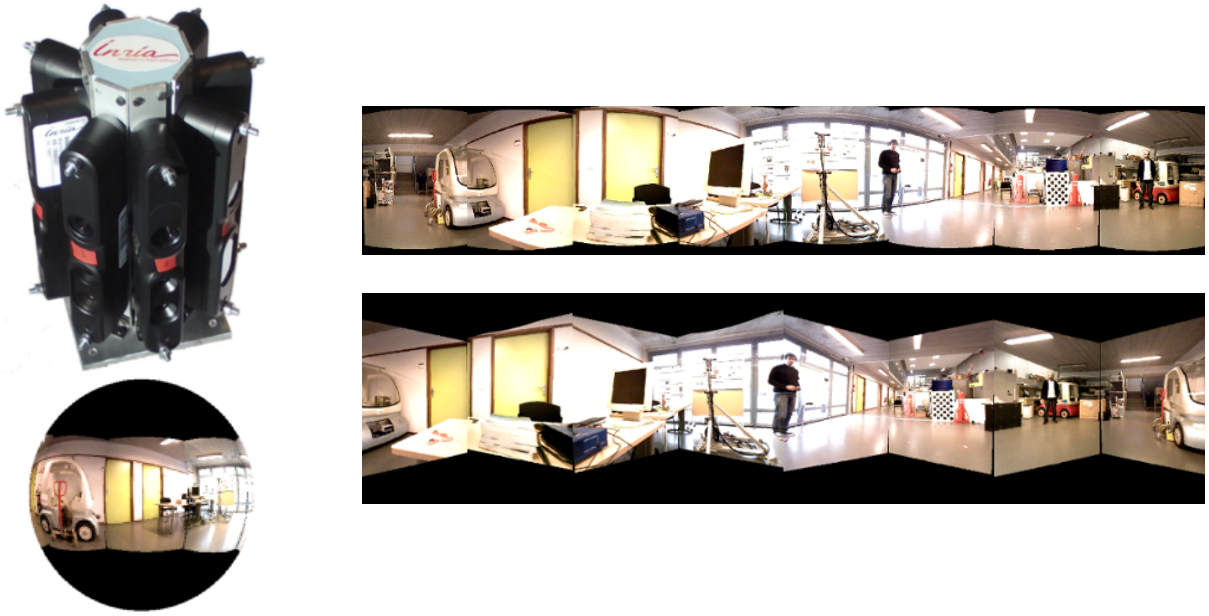

Figure 2.3 – Indoor RGB-D rig (upper left corner) and wide FOV image examples: the stereographic (lower left), equirectangular (upper right) and cube projection of the equirectangular spherical image (lower right). Note the distortions of straight lines and corners in the equirectangular image.

2.3.2 Depth Computation using Stereo

In the case of stereo cameras [2], two calibrated sensors are used and the computation can be divided in two main phases: image rectification and disparity computation. The reader can see [Hartley and Zisserman, 2003] (Chapters 8, 9 and 11) and [Faugeras et al., 2001] (Chapters 5 and 7) for more details and properties of two view geometry for perspective images. We remark that the fundamental matrix, essential matrix and homography are projective properties and therefore are valid to any single viewpoint (central) camera, as the considered perspective and panoramic images. Then, the rectification phase of the spherical panoramas is similar to perspective rectification. In this context, epipolar lines (from perspective imagery) correspond to great circles [3] in spherical vision. Therefore, the alignment of the poles of the spheres corresponds to the spherical rectification. This rectification can be done from the computation of the rotation matrix between the cameras coordinate systems, as the procedure described in [Gluckman et al., 1998] or in the appendix section of [Schönbein and Geiger, 2014].

From the rectified images, the disparity can be computed using dense stereo matching techniques as, for instance, the formulations in [Hirschmuller, 2008, Geiger et al., 2010, Yamaguchi et al., 2014]. This topic will be discussed in more detail in chapter 6.

2. Depth from stereo can be seen as a sub-problem of the dense visual SLAM, i.e., finding the structure given the relative pose.

3. Great circles are curves defined as the intersection of the sphere and a plane passing at the center of the sphere.

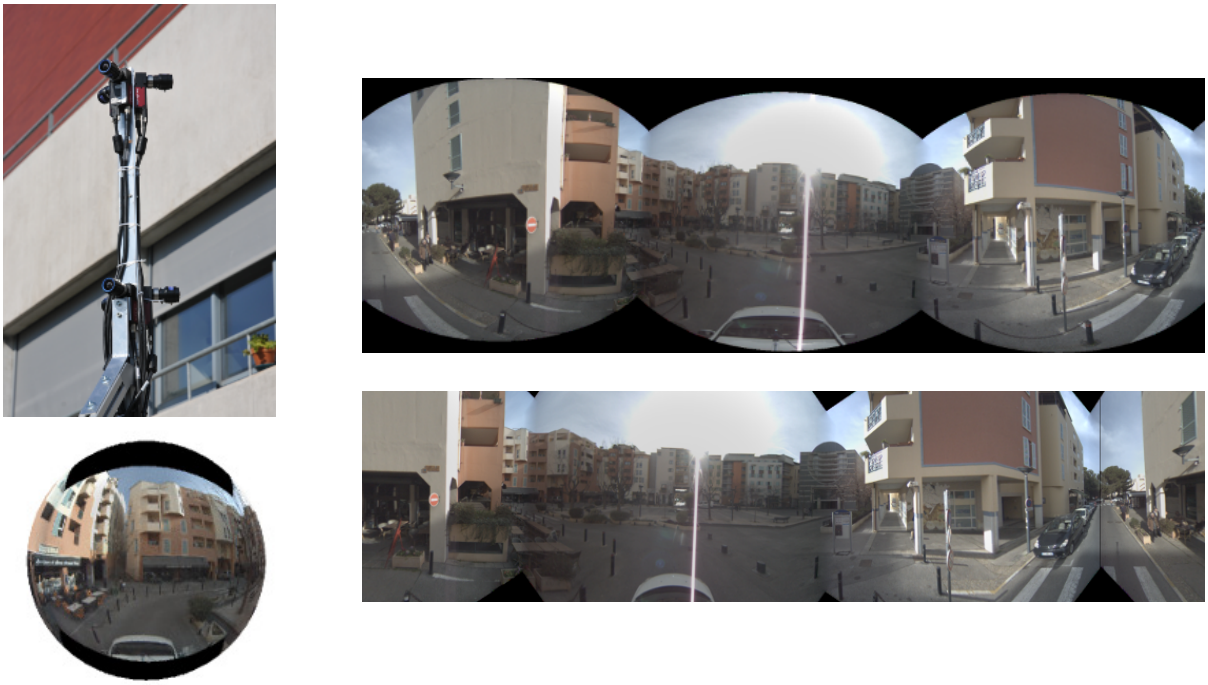

Figure 2.4 – Outdoor stereo rig (upper left corner) and wide FOV image examples: the stereographic (lower left), equirectangular (upper right) and cube projection of the equirectangular spherical image (lower right). Notice the shape distortion of the vehicle and of straight lines and corners in the equirectangular image.

2.3.3 Spherical RGB-D Acquisition Systems

For generating high resolution panoramic sequences with depth images, our research group has designed two novel customized devices: a device constituted by a rig of eight Asus Xtion Pro Live (Asus XPL) sensors for indoor scenes [Fernandez-Moral et al., 2014] and a stereo device composed of six cameras for outdoor scenes [Meilland et al., 2015] (see figs. 2.3 and 2.4). These two devices are assembled and calibrated such as to ensure, as much as possible, the central camera assumption. The intrinsic parameters $\mathbf{K}$ in (2.6) are therefore assumed to be known. Most of the image sequences used in this thesis are recorded at a frame rate of 20Hz, where the six global shutter cameras of the stereo system are synchronized, producing spherical images with a resolution of 1024x2048, which region-of-interest excluding the sphere poles is 664x2048. This defines a vertical FOV of approximately 120 degrees for the stereo rig. The resulting frames of the indoor rig have 960x1920 pixels, which region-of-interest of 320x1920 defines a vertical FOV of 60 degrees. We show in figs. 2.3 and 2.4 two frames for each respective sensor and in fig. 2.5 a full RGB-D frame with its normals. As it can be noticed, some non-modelled errors from the calibration and the stitching creates image discontinuities between edges in the scene and regions of the images with different exposures. These rigs were developed in the PhD theses of [Meilland, 2012, Fernandez-Moral, 2014] and were explored in the theses of [Chapoulie, 2012, Gokhool, 2015, Drouilly, 2015] for place recognition, mapping and semantic localization tasks.

2.4 Spherical Image Processing

Most traditional image processing tools were developed to perspective images, i.e., assuming flat images and with uniform spatial resolution. The spatial resolution of equirectangular panoramic images is, however, non-uniform and increases gradually towards the sphere poles. For instance, a rigorous interpolation in the sphere should consider the geodesic distance, as in the Slerp interpolation. Therefore, typical image processing tools such as the gradient and convolution related operators might be adapted to these images, as discussed in [Demonceaux et al., 2011, Bulow, 2002, Hadj-Abdelkader et al., 2008]. The first point of attention is to use the gradient in spherical coordinates. The gradient using the infinitesimal line length for the spherical coordinate system in (2.5) is:

$$\nabla_S(g) = \frac{1}{\rho\cos(\varphi)}\frac{\partial g}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{1}{\rho}\frac{\partial g}{\partial \varphi}\hat{\boldsymbol{\varphi}} + \frac{\partial g}{\partial \rho}\hat{\boldsymbol{\rho}} \tag{2.7}$$

which restricted to the unit sphere becomes $\nabla_S(g) = \left(\frac{\partial g}{\partial \varphi} \quad \frac{1}{\cos(\varphi)}\frac{\partial g}{\partial \theta}\right)^T$. As an illustration for the outdoor images with a vertical FOV of 120 degrees, the gradient using a classic Sobel filter needs to be scaled by a factor of two in the θ direction, for pixels in the periphery of the region-of-interest.

The size of the neighbourhoods to compute the gradient and other image processing operators need also to be redefined, as discussed in [Demonceaux et al., 2011, Daniilidis et al., 2002]. In this context, spherical harmonics are widely used for interpolation of general functions in the sphere [Bulow, 2002, Hadj-Abdelkader et al., 2008]. Unfortunately, spherical harmonics requires expensive computations of integrals using Legendre polynomials. In this work, we will consider the gradients in the sphere using eq. (2.7), but with a fixed neighbourhood size since the region-of-interest of the spherical frames is not located near the poles. Otherwise, adapted spatial filters should be preferred, as the ones presented in [Daniilidis et al., 2002, Bulow, 2002, Demonceaux et al., 2011].

2.5 Computation of Surface Normals

Due to their importance and vast application domain, normal/curvature estimation has been extensively investigated from different perspectives (e.g., [Mitra and Nguyen, 2003, Badino et al., 2011a, Jordan and Mordohai, 2014]). Good surveys about the choice of appropriate normal estimation techniques for ordered point clouds are presented in [Klasing et al., 2009, Badino et al., 2011a]. These works review and compare normal computation implementations from the perspective of efficiency and their robustness to increasing signal to noise ratio (SNR). It is worth noting that normals are related to "local" geometric properties which imply considering

suitable neighbourhoods. For instance, if the surface has sharp features then small neighbourhoods must be considered which increases noise influence. In fact, there is no simple trade-off since being robust to noise (increasing the size of neighbourhood) inevitably leads to smoothing (aliasing) scene details. [Hamann, 1993, Jordan and Mordohai, 2014] discussed the relationship between the neighbourhood size to the robustness to noise when using least squares fitting. A more complete analysis is done in [Mitra and Nguyen, 2003] by studying the effects of SNR, curvature, and sampling density and proposes some directions on how to automatically select the appropriate size of the neighbourhood for a better estimation using a plane fitting scheme. But the number of tuning parameters and the theoretical probability assumptions in the depth error make the applicability of their formulation limited. On the other hand, [Rusu et al., 2007] proposes an heuristic radius stability concept which adaptively search the good window neighbourhood size using k-means, i.e., no explicit error assumptions or error PDF estimation. Nevertheless being simpler than [Mitra and Nguyen, 2003], the efficiency is penalized. This algorithm is available in the PCL library.

In a first moment, we will privilege the efficiency and simplicity of the normal computation. Without loss of generality, let's assume a 3D surface $s : \mathbb{R}^3 \rightarrow \mathbb{R},\ s(\mathbf{P}) = 0$, passing through $\mathbf{P} \in \mathbb{R}^3$, smooth and with a normal existing everywhere. Local approximations of s are required when the surface is defined by a set of discrete measurements as, for instance, discrete point clouds or depth images. Parametric surface estimation algorithms perform regression to find the unknown surface $s(\mathbf{P})$ approximation, e.g., the coefficients of a plane [Hoppe et al., 1992] or a non-linear equation such as a quadric or cubic [Jordan and Mordohai, 2014]. A normal vector to this surface is represented as the orthonormal vector to the tangent plane Γ: $\mathbf{n}^T\mathbf{P} + d = 0$ and $d = -\mathbf{n}^T\mathbf{P}_0, \forall \mathbf{P}_0 \in \Gamma$. This concept of normal vector using a tangent planar patch leads to the widely known total least squares (TLS) algorithm [Hoppe et al., 1992, Badino et al., 2011a]. An even more efficient strategy than TLS is finite differences of three or more linearly independent 3D points. For example, the normal vector considering a local (3×3) window around the pixel $\mathbf{p}$:

$$\mathbf{n}(\mathbf{p}) = \mathbf{n}(u, v) = \|(\mathbf{P}(u+1, v) - \mathbf{P}(u-1, v)) \times (\mathbf{P}(u, v+1) - \mathbf{P}(u, v-1))\|_S, \tag{2.8}$$

which is very efficient but sensitive to noise. We remark that (2.8) has an equivalent expression in the sensor coordinate system. This allows the computation of the normals directly from depth images.

2.5.1 Normal Estimation in the Sensor's Domain

Given a surface defined implicitly as $s(x, y, z)$, the gradient is the direction of highest increase and therefore orthogonal to the level curves $s(x, y, z) = c$. Then, for each (x, y, z), ∇s is the normal to the surface $s(x, y, z) = 0$. For visualizing this, consider the 3D sphere in the origin

and with arbitrary radius ($r > 0$): $s(x, y, z) = x^2 + y^2 + z^2 - r^2$. We know that the normal to any point in the sphere is the viewing direction of the point, which is equal to the gradient $\nabla(s(x, y, z)) = (2x \ 2y \ 2z)^T$. Therefore, we just need to redefine the gradient from sensor coordinates (u, v, k) to spatial coordinates (x, y, z):

$$\nabla(g(u,v,k)) = \left(\frac{\partial g}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial x} + \frac{\partial g}{\partial k}\frac{\partial k}{\partial x}\right)\hat{\boldsymbol{x}} + \left(\frac{\partial g}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial y} + \frac{\partial g}{\partial k}\frac{\partial k}{\partial y}\right)\hat{\boldsymbol{y}} + (...)\hat{\boldsymbol{z}} \quad (2.9)$$

which depends on the sensor projection model. We develop this concept for the perspective and spherical depth images.

2.5.1.1 Perspective Depth Image

In the perspective case, the gradient operator from image coordinates $\mathbf{p} = (u, v, k)$ to spatial coordinates is:

$$\begin{cases} u = fX/Z + c_u \\ v = fY/Z + c_v \\ k = Z - f \end{cases} \Rightarrow \nabla_{nP}(g) = \frac{f}{Z}\frac{\partial g}{\partial u}\hat{\boldsymbol{x}} + \frac{f}{Z}\frac{\partial g}{\partial v}\hat{\boldsymbol{y}} + \left(\frac{(c_u - u)}{Z}\frac{\partial g}{\partial u} + \frac{(c_v - v)}{Z}\frac{\partial g}{\partial v} + \frac{\partial g}{\partial k}\right)\hat{\boldsymbol{z}} \quad (2.10)$$

Therefore from the implicit surface representation $s = Z - \boldsymbol{\mathcal{D}}(\mathbf{p}) = 0$ and exploring the previous framework from section 2.5.1 and the spherical norm (2.4):

$$\mathbf{n}(\mathbf{p}) = \|\nabla_{nP}(s)\|_S = \left\| \left(\frac{-f}{\boldsymbol{\mathcal{D}}(\mathbf{p})}\frac{\partial \boldsymbol{\mathcal{D}}(\mathbf{p})}{\partial u} \quad \frac{-f}{\boldsymbol{\mathcal{D}}(\mathbf{p})}\frac{\partial \boldsymbol{\mathcal{D}}(\mathbf{p})}{\partial v} \quad 1 - \frac{(c_u-u)}{\boldsymbol{\mathcal{D}}(\mathbf{p})}\frac{\partial \boldsymbol{\mathcal{D}}(\mathbf{p})}{\partial u} - \frac{(c_v-v)}{\boldsymbol{\mathcal{D}}(\mathbf{p})}\frac{\partial \boldsymbol{\mathcal{D}}(\mathbf{p})}{\partial v} \right)^T \right\|_S . \quad (2.11)$$

2.5.1.2 Spherical Depth Image

Similarly to the perspective case, we need a gradient operator that maps spherical $\mathbf{p} = (\theta, \varphi, \rho)$ to scene Cartesian coordinates. This can be done using the previous analytic derivations as in [Badino et al., 2011a] or using a geometric construction. We will flavor here the alternative geometric formulation, since the geometric construction reduces the complexity and gives more flexibility for adapting the computation for different spherical coordinate systems than the one in eq. (2.5). Using the gradient in spherical coordinates (2.7), the normal $\mathbf{n}_S$ of $s = \rho - \boldsymbol{\mathcal{D}}(\mathbf{p}) = 0$ in the sensor coordinate system is:

$$\mathbf{n}_S(\mathbf{p}) = \|\nabla_S(s)\|_S = \left\| \left(\frac{-1}{\boldsymbol{\mathcal{D}}(\mathbf{p})\cos(\varphi)}\frac{\partial \boldsymbol{\mathcal{D}}(\mathbf{p})}{\partial \theta} \quad \frac{-1}{\boldsymbol{\mathcal{D}}(\mathbf{p})}\frac{\partial \boldsymbol{\mathcal{D}}(\mathbf{p})}{\partial \varphi} \quad 1 \right)^T \right\|_S \quad (2.12)$$

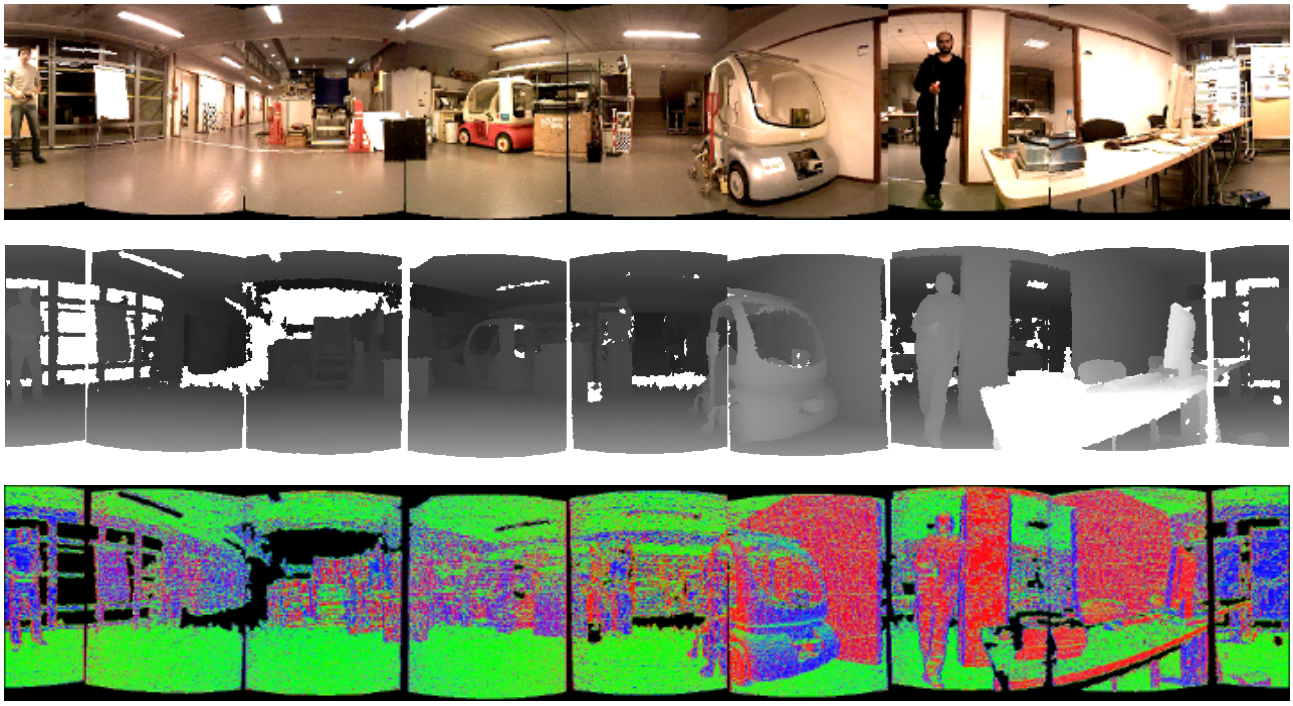

Figure 2.5 – RGB image (top), depth image (middle) and normal estimation (bottom) using a frame from the RGB-D indoor rig. The colors in the depth encode the inverse distance, where distant points from the camera have darker color. The colors in the normal image encode the orientation of the surfaces of the scene. As it can be noticed, the calibration of the rigs creates some discontinuities in the RGB and depth images. Furthermore, we can notice the depth errors inside smooth surfaces, as depicted by the changing colors of the normals in the walls, floor and ceiling.

The spatial coordinates are a rotated version of the spherical one and therefore:

$$\mathbf{n}(\mathbf{p}) = \mathbf{R}(\theta\mathbf{e_2})\mathbf{R}(-\varphi\mathbf{e_1})\mathbf{n}_S(\mathbf{p}) \tag{2.13}$$

with the rotation matrices around $\mathbf{e_2} = \hat{\mathbf{y}}$ and $\mathbf{e_1} = \hat{\mathbf{x}}$ axes, being computed using eq. (4.3). $\mathbf{R}(\theta\mathbf{e_2})\mathbf{R}(-\varphi\mathbf{e_1})$ can be computed only once and stored in a lookup table.

2.5.2 Normal Estimation Accuracy

The image gradients in (2.11) and (2.12) establish the level of detail and the sensitivity to noise of the normal computation. It is worth recalling that there is no simple trade-off since being robust to noise (increasing the size of neighbourhood) inevitably leads to smoothing details (as known as aliasing). For efficiency, we can use a classic centred finite differences kernel using a window of (3×3). Considering this kernel, the expressions in (2.11) and (2.12) are equivalent to eq. (2.8). Other standard gradient operators such as Sobel, Gaussian or Prewitt are equally valid. A discussion about the validity of this assumption is done later in

chapter 6. An example of the estimated normals is given in fig. 2.5 using the indoor sensor rig.

2.6 Summary and Closing Remarks

This chapter presented the basic tools and representation of perspective and spherical RGB-D images used along the thesis. We introduced the image projections and the panoramic spherical modelling. In the sequence, the RGB-D spherical indoor and outdoor rigs are presented, as well as the normal vector estimation. Finally, although working with spherical frames offers advantages, e.g., the better conditioning of mapping and ego-motion estimation, it also comes with intrinsic difficulties because most of the available optimization and image processing frameworks are conceived and valid to Euclidean/perspective spaces. For instance, even basic concepts such as spatial isotropic derivatives, neighbourhood and blurring are not trivial in the sphere (e.g., [Daniilidis et al., 2002, Bulow, 2002, Hadj-Abdelkader et al., 2008, Demonceaux et al., 2011]).

Chapter 3

Image Registration and Overview of Related Works

Contents

3.1 Introduction

The capabilities of building a representation of the environment, as well as estimating its relative position, are essential for robotic systems to interact and to evolve in the environment. These tasks, when treated simultaneously from images, are denoted as visual simultaneous localization and mapping (VSLAM) (e.g., [Engel et al., 2015, Mur-Artal et al., 2015, Whelan et al., 2015]) and structure from motion (SfM) (e.g., [Snavely et al., 2006, Wu, 2013]) problems tackled by the robotics and computer vision communities. Due to the wide field of applications, a vast (and rich) research literature exists around these topics. In this sense, the discussion in this chapter will be mainly restricted to robotic applications, where real-time performance is expected. This computation effort requirement restricts the use of bundle adjustment tools which are often applied in SfM [1]. Therefore, we describe mainly "sequential" estimation techniques for image registration and mapping. Good introductory texts (but not restricted to) are computer vision textbooks [Hartley and Zisserman, 2003, Faugeras et al., 2001], [Scaramuzza and Fraundorfer, 2011] about visual odometry and state-of-the-art registration algorithms, [Marchand et al., 2016] with an overview of pose estimation/tracking in virtual reality applications and, [Scharstein and Szeliski, 2002, Snavely et al., 2006] for mapping from images and the references included therein.

An essential block of most VSLAM systems is camera tracking, i.e., determining the location of the camera along the image sequence. Sequential vision-based registration methods track pixels between subsequent image frames and are often denoted as visual odometry (VO) when the tracking is also used to recover the camera motion over time. In short, most registration methods can be grouped following two categories: monocular/stereo and feature/appearance-based. In this chapter, we give further details about these categories. In the next sections, we expand and relate these concepts of registration and mapping from images to robotic applications, i.e., taking also into account the computational effort involved in the proposed algorithms. Moreover, we aim to position these algorithms (when appropriate) to the following challenging conditions such as:

— Large motions. Wide baseline images are prone to occlusions and non modeled appearance changes. In particular, direct registration methods assume local linearized approximations of the pixel appearance, which might be invalid in cases of large displacements.

— Model complexity and storage capacity. The complexity and size of the considered scene models can restrict the application of some algorithms to limited or relatively small scales.

— Textureless regions. Images acquired from weak-textured scenes are prone to noise when computing either the scene structure from stereo or the relative movement between the

1. A state of the art example is the tool chain of combining visualSfM and dense multiple view stereo (MVS) [Wu, 2013].

images using RGB registration.

The rest of this chapter is organized as follows. In section 3.2, we introduce image registration related works and perform a brief taxonomy of these methods in four different categories. Section 3.3 describes the framework used for the estimation of motion with direct registration techniques. In section 3.4, we discuss the convergence issues and some solutions proposed by the robotics and computer vision communities. Finally, section 3.5 summarizes the chapter and introduces the problems treated in the next two parts of the manuscript.

3.2 Image Registration

As described in section 3.1, the first registration category relates the number of cameras, with known relative pose, used in the tracking: a single camera (or monocular, e.g., [Silveira, 2014, Heng and Choi, 2016, Zhang et al., 2016]) and two or more cameras (stereo, e.g., [Morency and Darrell, 2002, Nistér et al., 2004, Comport et al., 2010, Tykkala et al., 2011, Engel et al., 2015]). Considering the monocular case, if the camera parameters are known, i.e. calibrated, the motion between the frames can be inferred up to a scale factor and then be sequentially integrated to compose the camera trajectory [2]. Stereo vision algorithms have the advantage, among others, of solving this motion-structure scale ambiguity inherent to monocular vision.

The second category relates how the pixel correspondences between the images is done. Feature-based methods can be decomposed, in general, as a sequential pipeline, starting by the extraction and matching of a subset of distinctive pixels (features) between the frames. Conversely, direct [3] (appearance-based) methods do not perform feature extraction and matching in a separate thread (see fig. 3.1) but minimizes an appearance error between the frames. In the next subsections, we detail the feature/direct and stereo categories.

3.2.1 Feature-Based Image Alignment

Feature-based methods (e.g. [Davison and Murray, 2002, Nistér et al., 2004, Kitt et al., 2010, Dryanovski et al., 2013]) rely on an intermediary extraction process based on thresholding [Harris and Stephens, 1988, Lowe, 2004], before matching the features and recovering the camera motion. The features at each frame can be extracted, for instance, using a classic difference of Gaussian (DoG) filter, Harris corner detector [Harris and Stephens, 1988] or the more recent FAST [Rosten et al., 2010]. Because of its high repeatability and low computational cost, FAST is one of the most popular feature detectors for real-time applications. In the subsequent stage of the pipeline, descriptors are computed for each detected feature (e.g., SIFT [Lowe, 2004],

2. Please refer to [Zhang, 1997, Malis and Vargas, 2007] about the extraction of the motion from essential and homography matrices and [Strasdat et al., 2010] for details about how to ensure the consistency of the scale factor during the trajectory estimation.

3. The term template-based is equally employed to refer to direct approaches.

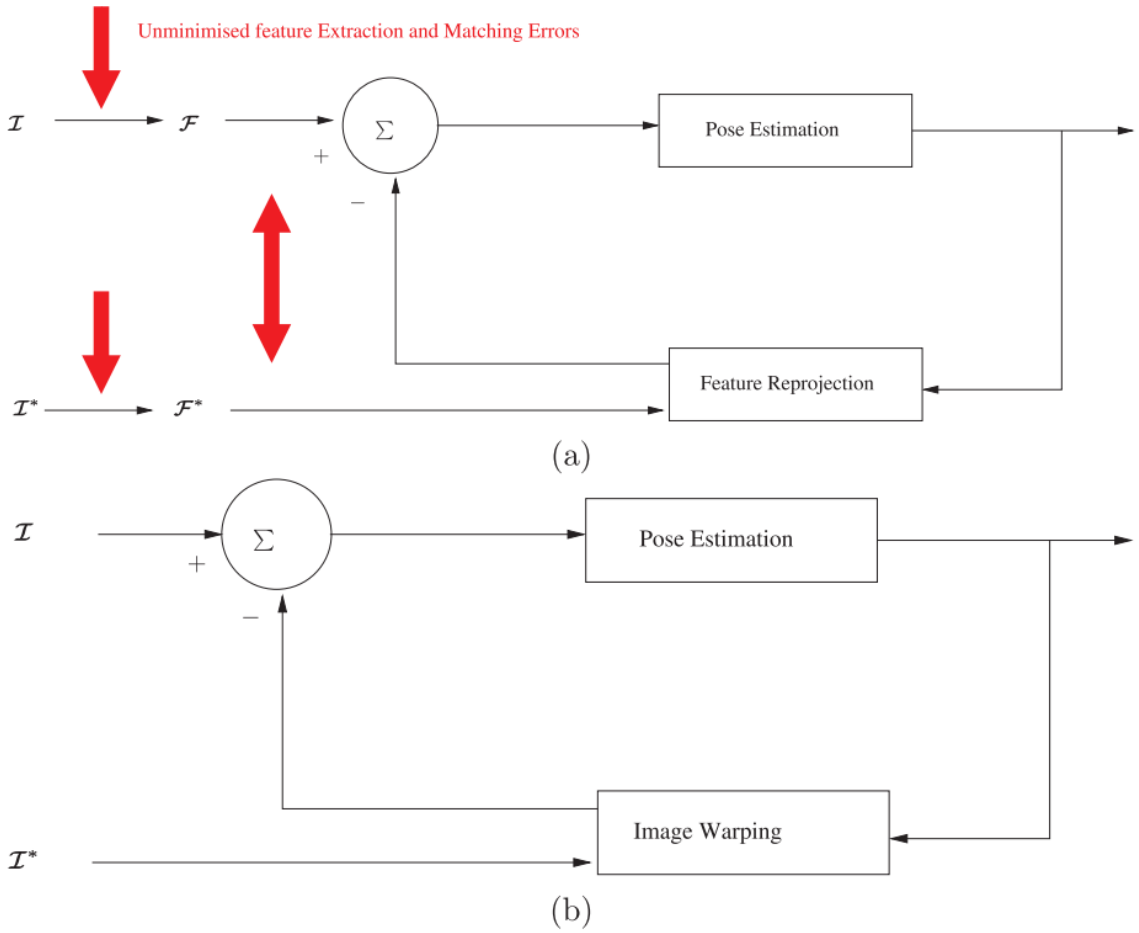

Figure 3.1 – Feature-based vs direct-based registration methods pipeline. The red arrows in the top scheme are the intermediary feature extraction from the images (denoted as $\mathcal{F}^*$ and $\mathcal{F}$) and their matching. Conversley, direct methods use the pixels intensities in order to register the frames, without any feature extraction or matching. Image courtesy of [Comport et al., 2010].

BRIEF [Calonder et al., 2010], ORB [Rublee et al., 2011, Mur-Artal et al., 2015] and then matched using a metric such as the Euclidean or hamming distances.

After the extraction and matching of the features, the tracking is done using the 8-point algorithm (monocular uncalibrated) [Hartley, 1997], 5-point algorithm (monocular calibrated) [Nistér et al., 2004] or an SVD (stereo) [Horn and Schunck, 1981, Nistér et al., 2004] (please refer to fig. 3.2 for some applications resulted from feature-based registration). The feature extraction and matching process is often noisy and not robust to outliers, and therefore it generally relies on higher level robust estimation techniques and on filtering to increase the tracking accuracy (e.g., [Nistér et al., 2004] or [Buczko and Willert, 2016] combining RANSAC with an adapted outlier rejection). An schematic of the stages of feature-based techniques can be seen in fig. 3.1 upper image. Finally, it is worth noting that implementations in different languages (Matlab, C++, Octave) of feature detectors, descriptors and trackers are available

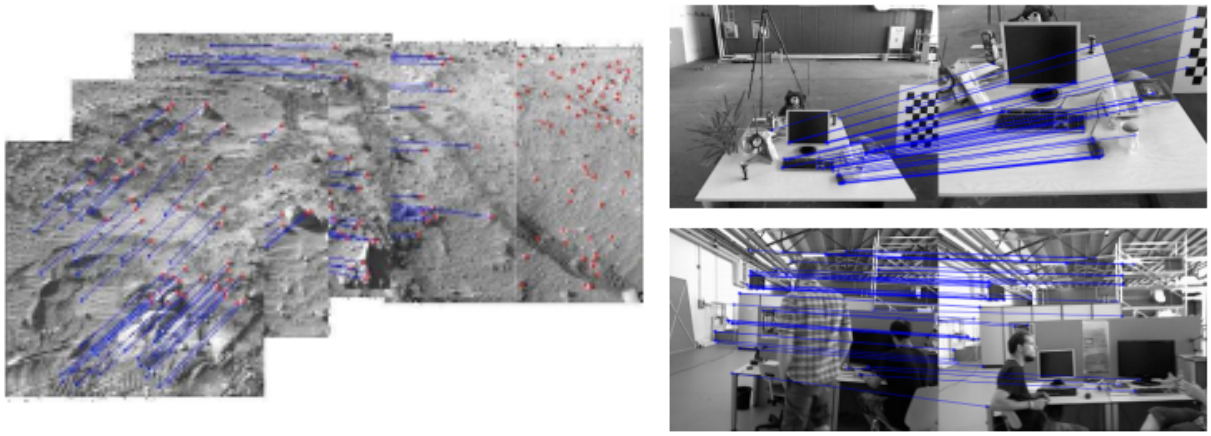

Figure 3.2 – Feature-based registration examples. The left figure depicts a stereo registration computed by NASA's exploration rover in Mars. The right plot shows feature correspondences using the ORB-SLAM algorithm in challenging apperance and scale conditions. Images courtesy of [Maimone et al., 2007] and [Mur-Artal et al., 2015] respectively.

in open source libraries such as, for instance, OpenCV tracking API or the VLFeat library [Vedaldi and Fulkerson, 2008].

3.2.2 Direct Camera Tracking

Conversely to feature-based methods, direct (appearance-based) approaches do not perform feature extraction and matching in a separate thread [Lucas and Kanade, 1981, Irani and Anandan, 2000]. All the image content is used and the matching is implicitly encoded in the minimization of an appearance error. The camera motion is estimated by minimizing a nonlinear error between images via a parametric warping function. In this way, the matching and the tracking are performed simultaneously at each step of the optimization (see fig. 3.1 lower image scheme and the left images of fig. 3.3). Direct approaches are closely related to dense optical/scene flow estimation problems. Briefly, optical flow is the apparent motion of pixels between subsequent images induced by a motion in the Euclidean scene space, which is under-constrained since the displacement of each pixel has two unknowns [4]. To handle this problem, the seminal work of Lucas-Kanade [Lucas and Kanade, 1981, Baker and Matthews, 2006] assumed a similar flow in a neighbourhood of a subset of distinctive pixels. This simplification reduces the number of unknowns to a sufficient number of linear equations to solve the problem, although the approach is somewhat limited to the neighbourhood constancy assumption, i.e., a local method [5]. This assumption is also implicitly considered in direct image registration

4. The general optical flow (tracking) problem can be stated as determining for each pixel $\mathbf{p}$ the displacements $(\mathbf{U}(\mathbf{p}), \mathbf{V}(\mathbf{p})) \in \mathbb{R}^2$ that makes the correspondence of pixels between two images.

5. Conversely to "local" flow formulations, "global" flow estimation techniques compute the flow using energy costs considering all pixels, (e.g., [Horn and Schunck, 1981, Dosovitskiy et al., 2015, Jaimez et al., 2017]). However, these energies use a first order Taylor expansion of the pixel intensity, i.e., are local in the sense of the motion. The main drawback of "global" flow approaches is their computational cost, the best cases with a

techniques.

Classically, direct approaches have focused on region-of-interest tracking whether they are modeled by affine (e.g., [Hager and Belhumeur, 1998]), planar (e.g., [Lucas and Kanade, 1981, Baker and Matthews, 2001, Silveira, 2014]), multiple-plane tracking (e.g., [Mei et al., 2006, Caron et al., 2011]) or trifocal/quadrifocal tensor and stereo (e.g., [Klose et al., 2013, Comport et al., 2010, Florez et al., 2012]). Again, the tracking is done from the minimization of a dissimilarity metric such as, for instance, the sum of squared differences (SSD) (e.g., [Comport et al., 2010]), normalized cross-correlation (NCC) (e.g., [Scandaroli et al., 2012]), the mutual information (e.g., [Dame and Marchand, 2010]) or the structural similarity (SSIM) (e.g., [Singh et al., 2017]). The choice of the appearance metric conditions the robustness to varying situations, such as, illumination changes [Scandaroli et al., 2012, Alismail et al., 2016, Singh et al., 2017]. Once the warping function parameters are estimated, the motion can be inferred from the decomposition of homography or essential matrices (e.g., using [Zhang, 1997, Malis and Vargas, 2007]). In [Comport et al., 2007, Comport et al., 2010] direct approaches were generalized to track the camera six DOF pose using stereo.

3.2.3 RGB-D Registration and Mapping

The recent advent of low-cost commodity RGB-D sensors, such as the Microsoft's Kinect or Asus's Xtion Pro, has drastically boosted the development of stereo registration and mapping techniques (e.g., the RGB-D registration and VSLAM systems presented in [Henry et al., 2014, Newcombe et al., 2011a, Kerl et al., 2013a, Meilland and Comport, 2013, Korn et al., 2014, Dryanovski et al., 2013, Gutierrez-Gomez et al., 2016, Kerl et al., 2015]). These commercial sensors are integrated with two infrared sensors to triangulate the objects distances, and therefore avoiding the matching of pixels from two images [6]. We will refer to stereo techniques simply as RGB-D, although the process and noise affecting the depth from stereo and from RGB-D devices have different characteristics.

[Dryanovski et al., 2013] presented a visual odometry and mapping algorithm relying on sparse image features (feature-based). The registration is done with a point-to-point iterative closest point (ICP) [Pomerleau et al., 2015] between the incoming RGB-D images and a global feature map stored in a Kalman filter. However, the storage capacity and computational burden restrict their algorithm to work in limited scene scales. One interesting aspect of their formulation is the uncertainty modeling of the depth images that can detect possible "flying pixels" [Wasenmüller and Stricker, 2016]. The depth uncertainty is modeled with a Gaussian mixture process for capturing occlusions and problematic features, such as object edges (which cannot be predicted using traditional uncertainty models such as, for instance, [Khoshelham and Elberink, 2012, Wasenmüller and Stricker, 2016]). [Newcombe et al., 2011a] fused RGB-D

running time in the order of seconds for low resolution images [Dosovitskiy et al., 2015].

6. The infrared limits the use of these sensors to indoor scenes with moderate lighting conditions.

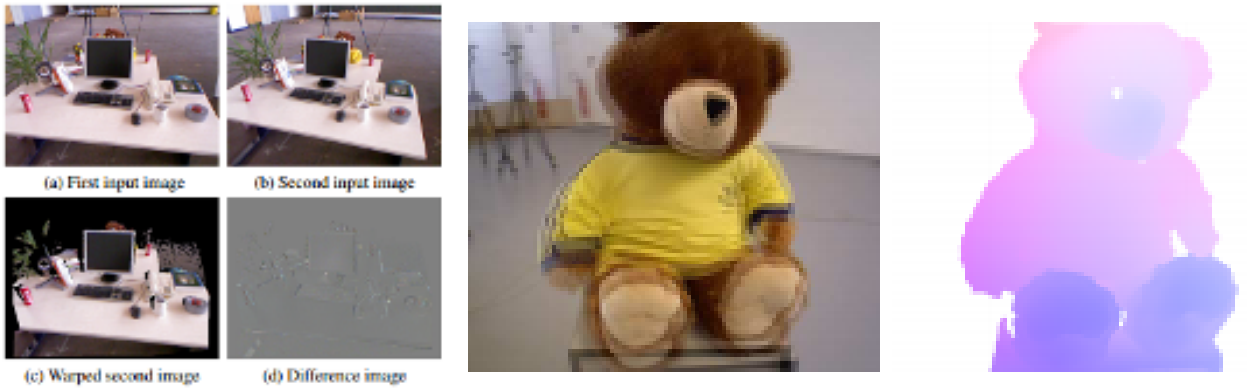

Figure 3.3 – Direct RGB-D registration computation (left images) and scene flow example using the TUM freiburg3_ teddy sequence. The camera motion is computed using the difference of pixel intensities in the direct RGB-D registration (left figure). The scene flow (center and right figures) encodes the 3D displacement vectors of the points using a direct registration procedure. Images courtesy of [Kerl et al., 2015] and [Quiroga et al., 2014] respectively.

frames in a voxel-based representation and perform the tracking between individual frames and the fused/accumulated model using a direct method. Similarly to [Dryanovski et al., 2013], this technique is restricted to relatively small scenes.

[Tykkala et al., 2011] was one of the pioneers to extend [Comport et al., 2010] to RGB-D sensors. Their formulation minimized jointly the geometric and photometric errors in the registration using a constant scaling factor based on the median values between the depth and intensity images. [Kerl et al., 2013a] proposed a keyframe-based RGB-D SLAM method where the pose between the individual frames is recovered from a direct RGB-D registration similar to [Tykkala et al., 2011]. The scaling in this case is iteratively computed to normalize the distributions of the intensity and geometric errors. To overcome the storage issue, the authors explore the notion of compact mapping by storing only representative frames (as known as keyframes [Gokhool et al., 2015]) and therefore reducing the computational burden. [Quiroga et al., 2014, Jaimez et al., 2017] present a framework to jointly estimate the camera motion and the scene flow of rigid objects using RGB-D images (see fig. 3.3 for examples). The core of both methods is the segmentation of the scene in local rigid clusters and performing spatial and temporal predictions of the segmented objects between subsequent frames. Combining the segmentation with spatial and temporal regularization allows a more accurate flow computation and a more accurate camera motion estimation. The flow computation in both [Quiroga et al., 2014, Jaimez et al., 2017] is more efficient than other state-of-the-art energy techniques (e.g., [Weinzaepfel et al., 2013]). However, these approaches depend on a large extend to a correct segmentation of the scene and the tuning of the temporal prediction and regularization parameters.

A simplified characterization of the described registration categories are depicted in Table 3.1, showing some characteristics of direct and feature-based camera tracking techniques for

monocular and RGB-D settings.

3.3 Direct Image Registration Framework

In this thesis, we focus on recovering the relative pose using parametric direct methods, i.e., using the pixel intensities of the images and without performing feature extraction and matching[7]. Consider a pair of image frames $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{S}}^*$ containing the respective intensity and depth images. Our main goal is to estimate the relative position and orientation between the frames, represented by the matrix $\mathbf{T}$, that minimizes an appearance error metric f (the dissimilarity) between the intensity and depth images:

$$\mathbf{e}(\mathbf{p}, \mathbf{T}) = f(\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \boldsymbol{\mathcal{D}}(\mathbf{p}), \mathbf{T})), \boldsymbol{\mathcal{S}}^*(\mathbf{p})) \tag{3.1}$$

where w is a warping function relating corresponding pixel positions in the frames. The relative pose $\mathbf{T}$, with six degrees of freedom, is represented throughout the manuscript using the matrix form in the special Euclidean group:

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{(1\times 3)} & 1 \end{pmatrix} \in \mathbb{SE}(3) \tag{3.2}$$

where $\mathbf{R} \in \mathbb{SO}(3)$ is the rotation matrix (encoding the orientation) and $\mathbf{t} \in \mathbb{R}^3$ the translation (encoding the position). For the optimization of the error (3.1), we will introduce in section 3.3.3.1 an intermediary representation using the instantaneous angular and linear velocities of the camera. An RGB-D registration scheme is shown in fig. 3.4 for a perspective camera. We present in the next sections the technical framework to find the pose minimizing the appearance error between the images $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{S}}^*$.

7. Although there is no explicit matching as in feature-based registration, an implicit matching is done in direct methods from the interpolation.

Table 3.1 – Resumed characteristics of image registration categories.

	Mono Feature	RGB-D Feature	Mono Direct	RGB-D Direct
Accuracy	•	•	•••	•••
Convergence domain	•••	•••	•	••
Scale ambiguity	Yes	No	Yes	No
Photometric model	NA	NA	Textured scenes	General scenes
Geometric model	NA	NA	Planar/simple scenes	General scenes
Computational effort	•	•	••	•••

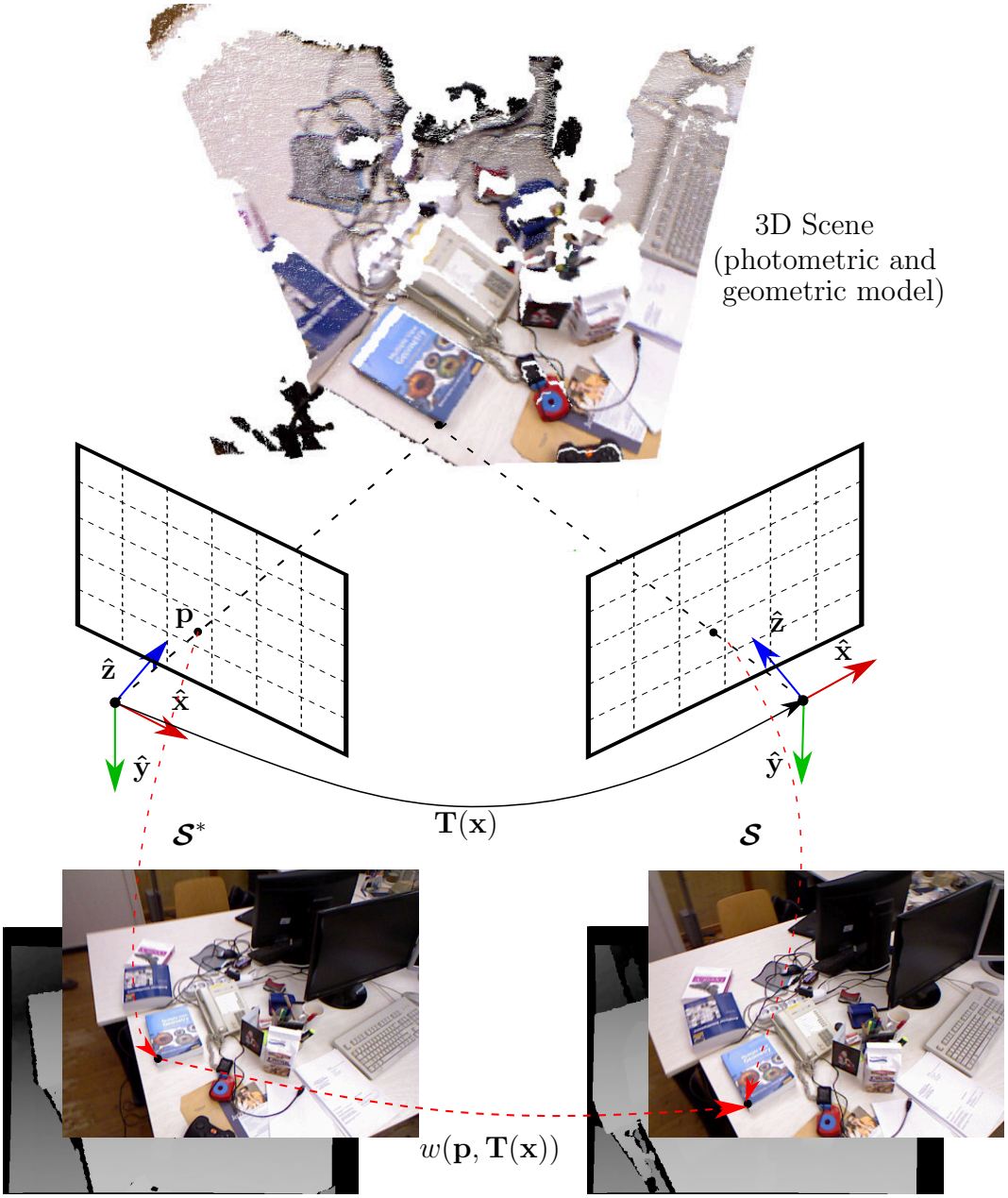

Figure 3.4 – Schematic of projections and RGB-D image registration using two frames from the TUM fr1/xyz RGB-D dataset.

3.3.1 Warping and Virtual Views

The first stage to solve the problem in eq. (3.1) is to consider the inverse situation, i.e., assuming the pose $\mathbf{T}$ known and to analyze how pixel correspondences is done. The point correspondences between frames are modeled by the warping function w under co-visibility conditions from different viewpoints. While a large number of warping functions can be proposed, e.g., using linear, affine or homography (please refer to [Baker and Matthews, 2006] for an overview of general warps), knowing an estimate of the scene structure allows us to use an efficient 3D warping $w : \mathbb{P}^2 \times \mathbb{R}_+ \times \mathbb{SE}(3) \rightarrow \mathbb{P}^2$, employing directly the pose $\mathbf{T}$ between the

viewpoints, as depicted in fig. 3.4. Denoting $\mathbf{K}$ the intrinsic camera matrix (2.2) and Π_S being the spherical projection (2.6), the corresponding warping functions are given by:

$$\begin{aligned} &\bullet \text{ Perspective: } w(\mathbf{p},\boldsymbol{\mathcal{D}}(\mathbf{p}),\mathbf{T}) = \|\boldsymbol{\mathcal{D}}(\mathbf{p})\mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{p} + \mathbf{K}\mathbf{t}\|_P \\ &\bullet \text{ Spherical: } w(\mathbf{p},\boldsymbol{\mathcal{D}}(\mathbf{p}),\mathbf{T}) = \Pi_S\left(\|\boldsymbol{\mathcal{D}}(\mathbf{p})\mathbf{R}\Pi_S^{-1}(\mathbf{p}) + \mathbf{t}\|_S\right) \end{aligned} \tag{3.3}$$

For notation compactness we denote $w(\mathbf{p},\mathbf{T}) := w(\mathbf{p},\boldsymbol{\mathcal{D}}(\mathbf{p}),\mathbf{T})$. It is worth noting that the aforementioned 3D warping (3.3) considers the same camera model (spherical to spherical and perspective to perspective). Of course, these expressions can be easily extended to any hybrid warping, e.g., perspective to spherical, spherical to perspective and so forth.

3.3.1.1 Rendering Virtual Frames

The point correspondence in eq. (3.3) allows the rendering of virtual images in a local domain of a frame $\boldsymbol{\mathcal{S}}^*$, encoding the photometric and geometric models of the scene. The complexity of the scene models increases as we consider effects such as occlusions, lighting conditions or viewpoint into account. However, we can simplify the image warping by considering stationary models in a local domain. For the geometric model, we can assume static scenes with minimal occlusions from the viewpoint related by the pose $\mathbf{T}$:

$$\begin{aligned} &\bullet \text{ Perspective: } \boldsymbol{\mathcal{D}}(w(\mathbf{p},\mathbf{T})) = \mathbf{e_3}^T(\mathbf{R}\mathbf{K}^{-1}(\mathbf{p})\boldsymbol{\mathcal{D}}^*(\mathbf{p}) + \mathbf{t}) \\ &\bullet \text{ Spherical: } \boldsymbol{\mathcal{D}}(w(\mathbf{p},\mathbf{T})) = \|\mathbf{R}\Pi^{-1}(\mathbf{p})\boldsymbol{\mathcal{D}}^*(\mathbf{p}) + \mathbf{t}\|_2. \end{aligned} \tag{3.4}$$

For the photometric model, considering the scene composed of Lambertian surfaces [8], constant lighting conditions [9] and a continuous geometric model in this local domain, i.e., the sensors sampling and surface occlusions can be neglected as in (3.4). Consequently, the intensity value in the virtual image can be simplified to:

$$\boldsymbol{\mathcal{I}}(w(\mathbf{p},\mathbf{T})) = \boldsymbol{\mathcal{I}}^*(\mathbf{p}). \tag{3.5}$$

The warping function (3.3) in (3.4) and (3.5) often generates non-integer pixel positions in the virtual image. Therefore, we must resort to image interpolation techniques for computing the image values in (3.4) and (3.5) from $\boldsymbol{\mathcal{D}}^*$ and $\boldsymbol{\mathcal{I}}^*$. Classic interpolation techniques are the nearest neighbour, bilinear, bicubic or cubic B-Spline. The bilinear interpolation is chosen for

8. Lambertian surfaces maintain their brightness appearance independent of the viewing direction.
9. This photometric model can be easily extended to consider more complex illumination models, e.g., an affine illumination $\boldsymbol{\mathcal{I}}(w(\mathbf{p},\mathbf{T})) = \alpha\boldsymbol{\mathcal{I}}^*(\mathbf{p}) + \beta$, with (α,β) constants in a window around the pixel $\mathbf{p}$.

Figure 3.5 – Virtual image rendering example using the spherical warping at a different viewpoint. The virtual view, shown in the second row, is rendered using the current frame (first image) at the the position of the reference frame (third row).

having the best trade-off between precision and computational effort compared to aforementioned techniques [Han, 2013] and since the region-of-interest of the spherical images are not near the poles. Finally, rendering the virtual image is equivalent to perform the warping and interpolation for all the pixels. These virtual images will be used to compute the errors between the reference and current frames to find their relative pose. An example of a "virtual frame" is shown the second row of fig. 3.5, where the virtual image is build from the warp of the current frame (first row) in the position of the reference frame (third row).

3.3.2 Recovering Motion From Images

As discussed in section 3.2.2, direct registration is closely related to dense optical flow estimation, relying the temporal/spatial continuity of apparent motion of pixels between subsequent images [Baker and Matthews, 2001]. We start by considering stationary photometric and geometric models of the scene, i.e., the frames appearance is invariant in time $\frac{\partial \boldsymbol{\mathcal{S}}(\mathbf{p})}{\partial t} = 0$. Under this assumption, the appearance change of the frame content can be approximated by a truncated Taylor expansion in a spatial local neighborhood $\mathbf{T} = \hat{\mathbf{T}}\mathbf{T}(\mathbf{x})$, such as:

$$\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) = \boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}})) + \nabla_{\mathbf{x}}(\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))))\Big|_{\mathbf{x}=\mathbf{0}_{6\times 1}} \mathbf{x} + \mathcal{O}(\|\mathbf{x}\|_2^2) \tag{3.6}$$

where $\hat{\mathbf{T}}$ is a pose guess and $\nabla_{\mathbf{x}}$ is the gradient of the frame to the parameter $\mathbf{x}$. The relation (3.6) defines the appearance of any virtual frame $\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x})))$ in a local neighborhood of the pose $\hat{\mathbf{T}}$, ideally with Lambertian surfaces, constant lighting conditions, static scenes and small motions. As discussed in section 3.3.1, the virtual frame $\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}}))$ in eq. (3.6) can be seen as a mapping of the position of the pixels (warping) of the reference frame $\boldsymbol{\mathcal{S}}^*$, i.e., using the virtual intensity and depth images as shown in eqs. (3.4) and (3.5):

$$\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}})) = \boldsymbol{\mathcal{S}}^*(\mathbf{p}). \tag{3.7}$$

We can particularize the direct approaches from the type of images being used, in the following four cases:

- Monocular or intensity only registration: $\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{I}}^*\}$ and $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}\}$. In this case only the intensity images are considered (non-metric information) such as [Silveira, 2014, Heng and Choi, 2016, Zhang et al., 2016]. These formulations suffers from the scene-motion scale ambiguity, justifying the inclusion of either IMU, laser or a stereo pair to condition the pose estimation.
- 3D – 3D registration: $\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{D}}^*\}$ and $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{D}}\}$. The cost function is composed only of depth information. The typical examples are direct ICP point-to-plane [Gelfand et al., 2003] and generalized ICP formulations [Korn et al., 2014, Pomerleau et al., 2015].
- Augmented intensity registration: $\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{D}}^*\}$ and $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}\}$. In this case, the depth is simply employed in the warping such as, for instance, the seminal work of [Comport et al., 2010] and [Morency and Darrell, 2002, Tykkala et al., 2011, Kerl et al., 2013a, Munoz and Comport, 2016a] with zero geometric scaling cost factor.
- RGB-D registration: $\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{D}}^*\}$ and $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}\}$. The cost function has both photometric and geometric error terms such as, for instance, [Morency and Darrell, 2002, Tykkala et al., 2011, Kerl et al., 2013a, Munoz and Comport, 2016a].

In this thesis, we are interested in 3D general warpings, i.e., the last three cases. In special, we will focus in the fourth case (RGB-D frames), since it retains the accuracy of intensity only approaches and increases the flexibility of the later by relaxing the constraint of high textured images, because the camera can still be tracked with a collection of textureless planes in RGB-D images.

3.3.3 Appearance Cost Minimization

The remaining problem is to select a suitable cost of the appearance error in eq. (3.7) and the minimization framework. Classical examples of appearance errors are the sum of squared differences (SSD), normalized cross correlation or the mutual information. For simplicity, we start by developing the third case ($\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{D}}^*\}$ and $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}\}$). The fourth case ($\boldsymbol{\mathcal{S}}^* =$

$\{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{D}}^*\}$ and $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}\}$) will be described in detail in chapter 5. Consider the following robust cost:

$$C = \min_{\mathbf{x}} \sum_{\mathbf{p}} \rho(e(\mathbf{p}, \mathbf{x})) \tag{3.8}$$

where ρ is a M-estimator and

$$e(\mathbf{p}, \mathbf{x}) = f(\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))), \boldsymbol{\mathcal{S}}^*(\mathbf{p})) = \boldsymbol{\mathcal{I}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}) \tag{3.9}$$

is the appearance intensity error related to the pixel $\mathbf{p}$. This cost can be re-written in a SSD form using an iterative weighted least-squares with diagonal weighting matrix $\mathbf{W}$ [Zhang, 1995]:

$$C(\mathbf{x}) = \mathbf{e}(\mathbf{x})^T \mathbf{W} \mathbf{e}(\mathbf{x}) \tag{3.10}$$

where $e(\mathbf{p}, \mathbf{x})$ is a vector composed by the intensity errors for all pixels and

$$\mathbf{W} = \text{diag}(w_1, ..., w_k) \tag{3.11}$$

states the confidence of each measure using the M-estimator (e.g., Huber, Tukey, Cauchy). Considering a robust objective function in the cost has the advantage of rejecting outliers during the minimization.

3.3.3.1 Minimal Motion Parametrization

In order to perform the image alignment and recover the relative pose, we need to minimize the appearance error in eq. (3.10). Although the pose in matrix form has the advantages of being unique, not presenting singularities and with simple analytic properties, it is not a minimal representation (sixteen elements, whose nine for the rotation). Hence, finding the motion six DOF by minimizing a cost related to these sixteen elements does not guarantee that the computed pose belongs to $\mathbb{SE}(3)$ [10]. Therefore, the minimization is not done with the pose in matrix form, but rather considering an intermediary motion parametrization in the manifold:

$$\mathbf{x} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})^T \tag{3.12}$$

10. A possible strategy to overcome this issue is to perform a projection of the motion in $\mathbb{SE}(3)$. One such way is doing a Frobenius normalization of the rotation matrix, as described in the appendix section of [Zhang, 2000].

using the instantaneous linear ($\boldsymbol{\upsilon}$) and angular velocities ($\boldsymbol{\omega}$), during a time interval $(t_0, t_0+\delta t)$. This parameter vector relates to the rigid transform $\mathbf{T}(\mathbf{x})$ through the exponential map [11]:

$$\begin{aligned} \exp: \quad & \mathbb{R}^6 \rightarrow \mathfrak{se}(3) \rightarrow \mathbb{SE}(3) \\ & \mathbf{x} \mapsto \mathbf{A}(\mathbf{x}) \mapsto \mathbf{T}(\mathbf{x}) = \exp(\mathbf{A}(\mathbf{x})\delta t) \end{aligned} \tag{3.14}$$

where $\mathbf{A}$ can be written as a linear combination of the generators of the algebra $\mathfrak{se}(3)$:

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} \mathbf{S}(\boldsymbol{\omega}) & \boldsymbol{\upsilon} \\ \mathbf{0}_{(1\times 3)} & 0 \end{pmatrix} \tag{3.15}$$

which is the Lie algebra of $\mathbb{SE}(3)$ at the identity element and $\mathbf{S}(\boldsymbol{\omega})$ is the skew $\mathbb{R}^{3\times 3}$ matrix of the angular velocity $\boldsymbol{\omega}$. Selecting the duration of the time interval as $\delta t = 1$, the exponential mapping in (3.14) can be computed efficiently using a closed form solution using the Rodrigues' formula [Gallier and Xu, 2002, Gallier, 2000]:

$$\mathbf{T}(\mathbf{x}) = \exp(\mathbf{A}(\mathbf{x})) : \begin{cases} \mathbf{R} = \mathbf{I}_{(3\times 3)} + \sin(\Theta)\mathbf{S}(\mathbf{n}_\Theta) + (1-\cos(\Theta))\mathbf{S}(\mathbf{n}_\Theta)^2 \\ \mathbf{t} = \left(\mathbf{I}_{(3\times 3)} + \frac{(1-\cos(\Theta))}{\Theta}\mathbf{S}(\mathbf{n}_\Theta) + \frac{(\Theta-\sin(\Theta))}{\Theta}\mathbf{S}(\mathbf{n}_\Theta)^2\right)\boldsymbol{\upsilon} \end{cases} \tag{3.16}$$

with $\mathbf{I}$ the identity matrix, $\Theta = \|\boldsymbol{\omega}\|_2$ and $\mathbf{n}_\Theta = \|\boldsymbol{\omega}\|_S$ for $\Theta \neq 0$. If $\Theta = 0$ the axis of rotation is not determined and therefore can be chosen arbitrarily (e.g., $\mathbf{n}_\Theta = \mathbf{e_1}$). Similarly, the inverse mapping $\log : \mathbb{SE}(3) \rightarrow \mathfrak{se}(3)$ can be directly obtained from (3.16), being valid in a local neighborhood of the rotation (see [Gallier, 2000] chapter 14):

$$\mathbf{x} = \text{vex}(\log(\mathbf{T}(\mathbf{x}))) : \begin{cases} \boldsymbol{\omega} = \| \text{ vex}\left(\mathbf{R} - \mathbf{R}^T\right) \|_S \Theta \\ \boldsymbol{\upsilon} = \left(\mathbf{I}_{(3\times 3)} - \mathbf{S}(\boldsymbol{\omega}) + \left(\frac{\tan(\Theta/2)-\Theta/2}{\tan(\Theta/2)}\right)\mathbf{S}(\boldsymbol{\omega}/\Theta)^2\right)\mathbf{t} \end{cases} \tag{3.17}$$

for $\Theta = \|\boldsymbol{\omega}\|_2 < \pi$ and $2\cos(\Theta) = \text{tr}(\mathbf{R}) - 1$. The operator vex is an overloaded operator such that:

$$\text{vex}(\mathbf{S}(\boldsymbol{\omega})) = \boldsymbol{\omega} \quad \text{and} \quad \text{vex}(\mathbf{A}(\mathbf{x})) = \mathbf{x}.$$

When $\|\boldsymbol{\omega}\|_2 = \pi$, we need to find $\mathbf{S}(\mathbf{n}_\Theta)$ satisfying $\mathbf{I}_{(3\times 3)} + \mathbf{S}(\mathbf{n}_\Theta)^2 = \mathbf{R}$. As $\mathbf{S}(\mathbf{n}_\Theta)$ is a skew-symmetric (3×3) matrix, this amounts to solving a system with three unknowns.

11. The rigid transform $\mathbf{T}(\mathbf{x})$ encodes the pose evolution considering constant velocities during a time interval δt. The position of a point $\mathbf{P}(t) \in \mathbb{R}^3$ relative to a fixed reference world frame at instant t_0 can be stated as the kinematic model of linear and angular velocities:

$$\frac{d\mathbf{P}(t)}{dt} = \boldsymbol{\upsilon} + \boldsymbol{\omega} \times \mathbf{P}(t) = \boldsymbol{\upsilon} + \mathbf{S}(\boldsymbol{\omega})\mathbf{P}(t). \tag{3.13}$$

Therefore, the pose $\mathbf{T}(\mathbf{x})$ in (3.2) (3.10) (3.14) is the solution of the linear ODE in (3.13), such as $\mathbf{P}(t_0+\delta t) = \mathbf{R}\mathbf{P}(t_0) + \mathbf{t}$.

Finally, it is worth noting that any pose derivative related to each DOF, e.g. $\frac{\partial \mathbf{T}(\mathbf{x})}{\partial x_i} = \mathbf{A}_i$, is given simply by the generators of $\mathfrak{se}(3)$:

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{e}_1 \\ \mathbf{0}_{1\times3} & 0 \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{e}_2 \\ \mathbf{0}_{1\times3} & 0 \end{pmatrix}, \ \mathbf{A}_3 = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{e}_3 \\ \mathbf{0}_{1\times3} & 0 \end{pmatrix}$$
$$\mathbf{A}_4 = \begin{pmatrix} \mathbf{S}(\mathbf{e}_1) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{pmatrix}, \ \mathbf{A}_5 = \begin{pmatrix} \mathbf{S}(\mathbf{e}_2) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{pmatrix}, \text{ and } \mathbf{A}_6 = \begin{pmatrix} \mathbf{S}(\mathbf{e}_3) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{pmatrix}, \tag{3.18}$$

which greatly simplifies the minimization expressions (the Jacobians) in the motion computation, while simultaneously enforcing the pose in the special euclidean group.

3.3.3.2 Efficient Second Order Minimization

In general, the cost in eq. (3.10) is not globally convex, since the appearance error is nonlinear. Hence, the estimation of the optimal solution depends on the appearance error approximation and on the optimization method. Our goal is to find a minimum equal or nearby the global minimum of the cost in (3.10). Unfortunately, global optimization techniques are too computationally expensive to be used [Yang et al., 2016]. Therefore, gradient-based optimization methods have been widely employed in the minimization, as these techniques present a good trade off between region of convergence and computational cost.

Several gradient-based minimization algorithms are available, such as, the steepest descent, Gauss-Newton, Levenberg-Marquardt, Powell's "dogleg" or the Efficient Second Order Minimization (ESM). Please refer to the appendix section of [Hartley and Zisserman, 2003] and to [Benhimane and Malis, 2004, Baker and Matthews, 2006] for further details of these algorithms. We further describe the ESM method since it generalizes the Gauss-Newton and the inverse compositional formulations, while maintaining interesting convergence properties. The ESM considers a second order approximation of the error:

$$\mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0},\mathbf{x})\mathbf{x} + \mathcal{O}(\|\mathbf{x}\|_2^3) \tag{3.19}$$

where $\mathcal{O}(\|\mathbf{x}\|_2^3)$ are the terms with three or higher orders and,

$$\mathbf{J}(\mathbf{0}) = \nabla_{\mathbf{x}}(\mathbf{e}(\mathbf{x})) = \nabla_{\mathbf{x}}(\boldsymbol{\mathcal{S}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))))\Big|_{\mathbf{x}=\mathbf{0}_{6\times1}}$$

by using eq. (3.7). The matrix $\mathbf{M}$ using the Hessian can be written as:

$$\mathbf{M}(\mathbf{0},\mathbf{x}) = \nabla_{\mathbf{x}}(\mathbf{J}(\mathbf{x}))\Big|_{\mathbf{x}=\mathbf{0}_{6\times1}} \mathbf{x}.$$

Interestingly, the first order expansion of the Jacobian in the origin can be written as:

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathcal{O}(\|\mathbf{x}\|_2^2). \tag{3.20}$$

Using (3.19) and (3.20), we have:

$$\mathbf{e}(\mathbf{x}) \approx \mathbf{e}(\mathbf{0}) + \left(\frac{\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})}{2}\right)\mathbf{x}$$

where using $\mathbf{J}(\mathbf{x})\mathbf{x} = \mathbf{J}^*(\mathbf{0})\mathbf{x}$ and from the first order optimal condition of (3.10), $\frac{\partial C(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{0}$, we finally have:

$$\mathbf{J}_{esm} = \frac{\mathbf{J}(\mathbf{0}) + \mathbf{J}^*(\mathbf{0})}{2}. \tag{3.21}$$

The photometric Jacobians $\mathbf{J}^*$ and $\mathbf{J}$ (Nx6) are computed for the reference and current frames. Due to the warping group properties, these can be decomposed for each pixel in:

$$\mathbf{J} = \nabla_{\mathbf{p}}(\boldsymbol{\mathcal{I}}(w(\mathbf{p}, \hat{\mathbf{T}})))\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{T}} \text{ and } \mathbf{J}^* = \nabla_{\mathbf{p}}(\boldsymbol{\mathcal{I}}^*(\mathbf{p}))\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{T}}, \tag{3.22}$$

with $\nabla_{\mathbf{p}}(\boldsymbol{\mathcal{I}})$ (1x3) as the image gradient. $\mathbf{J}_{\mathbf{w}}$ (3x12) is the Jacobian of the warping function which depends on the sensor projection model (e.g., perspective or spherical). $\mathbf{J}_{\mathbf{T}}$ (12x6) is the Jacobian of the pose related to the instantaneous angular and linear velocities ($\mathbf{x}$). Each row of this Jacobian corresponds to the flatten version of the generators $\mathbf{A}_i$ three first rows, as described in section 3.3.3.1 and in the appendix A. In summary, the cost in (3.10) is minimized iteratively through the update rule:

$$\boldsymbol{\mathcal{I}}_w(\mathbf{p}) = \boldsymbol{\mathcal{I}}(w(\mathbf{p}, \hat{\mathbf{T}})) \tag{3.23}$$

$$\mathbf{x} = -(\mathbf{J}_{esm}^T \mathbf{W} \mathbf{J}_{esm})^{-1} \mathbf{W} \mathbf{J}_{esm}^T (\boldsymbol{\mathcal{I}}_w - \boldsymbol{\mathcal{I}}^*) \tag{3.24}$$

$$\hat{\mathbf{T}} \leftarrow \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}). \tag{3.25}$$

The optimization stop conditions are defined by the maximum number of iterations and by a threshold of the increment in $\mathbf{x}$. The description of the Jacobians, the choice of the robust M-estimator and computation of the weights are described in detail in the appendix A.

3.4 Convergence of Registration Methods

The usage of direct registration approaches is increasing in the last decade because of the sub-pixel accuracy displayed in tracking applications [12] and because of the increase of available computation resources. However, these methods are not capable of handling wide baselines (displacements) between the frames. Among the strategies to increase the basin of convergence and convergence rate of direct methods, multi-resolution is widely employed due to its easy implementation and versatility (e.g., [Klose et al., 2013, Comport et al., 2010]). Multi-resolution consists on creating successive down-sampled images, creating a pyramidal structure. The optimization is done from the lowest resolution (top of the pyramid) to the highest image resolution level. The convergence of direct techniques is widely improved by considering multi-resolution schemes. In the same context, some state-of-the-art techniques, concerned with tracking and optical flow estimation [Hadj-Abdelkader et al., 2008, Brox and Malik, 2011, Muller et al., 2011, Braux-Zin et al., 2013] proposed a combination of feature and direct-based approaches for a trade off between accuracy and basin of convergence.

Some effort was also done in estimating "confidence" envelops where convergence is likely to happen, even though a general mathematical condition was not established for determining the convergence domain. Recently, [Churchill et al., 2015, Dequaire et al., 2016] described a framework for determining pose envelops for convergence of monocular feature-based registration (see fig. 3.6 for examples of convergence envelops in different scenes). The envelops encode the probability of convergence of the registration for different initial conditions. A Gaussian mixture process is used to estimate these envelops using the Teach and Repeat paradigm [Furgale and Barfoot, 2010]. The number of correct matched features is taken as index for the envelops inference. However, this space characterization needs an extensive exploration of the scene, which is quite prohibitive in most contexts.

Consequently, a central objective of this thesis is to investigate possible techniques to increase the basin of convergence and to propose adapted mapping techniques to efficiently represent the scene model, while maintaining nice convergence properties for direct techniques.

3.5 Summary and Closing Remarks

In this chapter, we presented an overview of sequential pose estimation methods from images. We introduced feature-based, direct (appearance-based) and the RGB-D image registration problems and presented some relevant solutions. Direct registration is performed by generating virtual image views using a pre-computed structure of the scene. Although these techniques are in general better conditioned and more accurate than feature-based approaches,

12. The sub-pixel accuracy in the image domain is not isometric to the motion accuracy of the camera in the Euclidean domain. Observe that a large motion can induce a slight change in the pixels coordinates for points far apart from the camera.

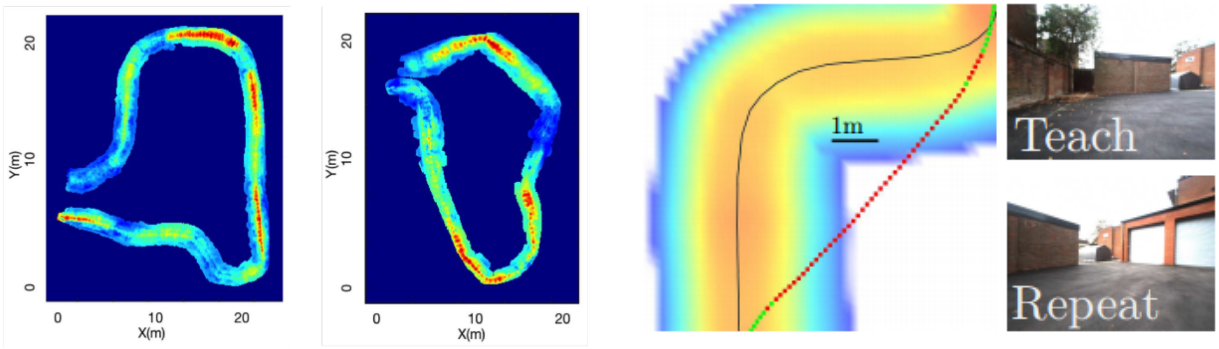

Figure 3.6 – Convergence envelops for image based localization. The envelops are built via the Teach and Reapeat paradigm using landmarks. Hotter colors indicate a high number of landmarks' matching than colder colors. As can be noticed, the convergence envelops vary drastically over the trajectory. First two images courtesy from [Dequaire et al., 2016] and last three images from [Churchill et al., 2015].

they display a reduced basin of convergence compared to feature-based techniques. Finally, we discussed the limitations of current state-of-the-art algorithms and solutions to improve the convergence of direct approaches. We recall that enlarging this domain is of practical interest not only for RGB-D sensors but, potentially, for all the visual odometry approaches based on direct registration.

In the subsequent chapters of this thesis, a particular attention is given to the topics of increasing this basin of convergence/the convergence rate and on building appropriate mapping representations. These are key aspects for generating valid scene models to be used later on in robot autonomous navigation or in scene rendering.

Part II

Direct RGB-D Registration with Large Motions

Introduction

As discussed in the introductory chapters, most parametric direct registration methods are based on the linearization of a warping function and, thus, the convergence is local. In this part, we describe two strategies to increase the basin of convergence of direct RGB-D tracking techniques. The first strategy, described in chapter 4, is a pose computation using the normals of the depth images with convenient properties. This pose computation is fast because of the characteristics of the normals and because it can exploit low-resolution depth images. Under certain configurations, the estimated motion converges in a single iteration to the solution of an ICP point-to-plane. Therefore, it can be used in an initialization procedure, as shown in fig. 3.7.

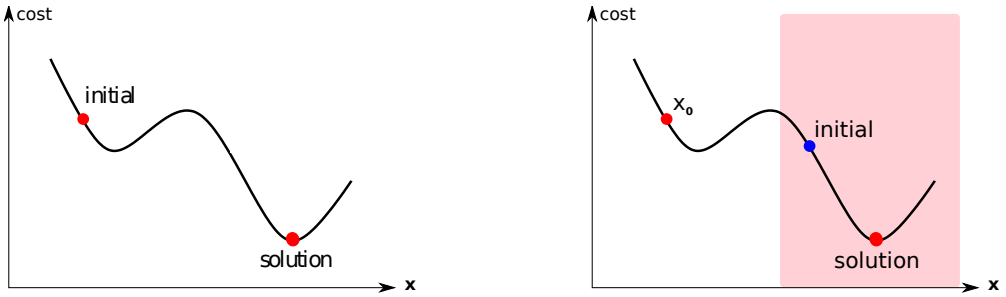

Figure 3.7 – Initialization formulation presented in chapter 4. The goal is to compute an initialization to the nonlinear cost (blue dot at the right graphic), allowing direct methods to work in the basin of convergence for pose refinement (the red shaded region).

The second strategy, described in chapter 5, is a formulation to update the weighting of the RGB-D error function along the optimization. The formulation uses the following reasoning. The intensity error has, in general, a very well defined minimum due to the alignment of boundaries of the intensity images. This explains in part the sub-pixel tracking accuracy displayed by direct methods. On the other hand, the geometric error is more flat in the neighborhood of the minimum, but it has a faster convergence when the motion is large. We describe an adaptive weighting technique based on these two observations, i.e., shaping the original cost, as shown in fig. 3.8.

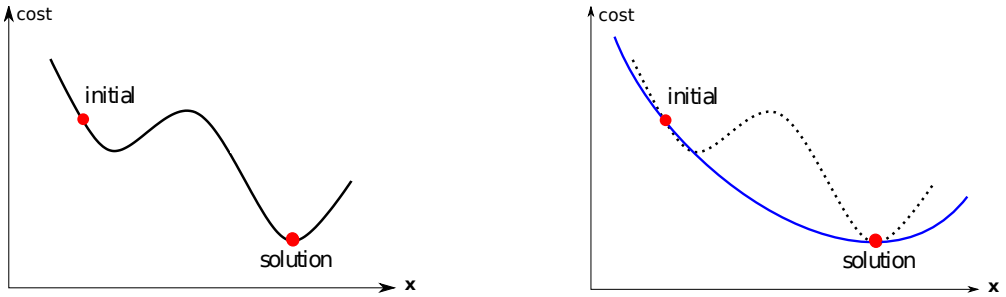

Figure 3.8 – The adaptive formulation presented in chapter 5. In this case, we want to shape the original cost function during the minimization, by balancing appropriately the photometric and geometric errors (blue curve at left).

Chapter 4

Efficient Pose Initialization from Normal Vectors

Contents

4.1 Introduction

The goal of registration techniques is to compute the motion, i.e., the relative pose from images or point clouds. In special, mobile robots require efficient registration algorithms, which are often iterative and assume a good pose initialization to converge. Pose initialization techniques are widely used in registration tasks such as using features [Hadj-Abdelkader et al., 2008] or by using external sensors such as inertial measurement systems [Kelly and Sukhatme, 2011] and wheel odometry [Maimone et al., 2007]. In this chapter, we present an efficient pose estimation method from the normal surface vectors of two depth images. Surprisingly, except in [Ma et al., 2016, Zhou et al., 2016a], the information gathered from normal vectors has been exploited mainly for outlier rejection in point cloud registration algorithms as showed, for instance, in the ICP survey in [Pomerleau et al., 2015] or in [Serafin and Grisetti, 2015]. Our formulation uses low-resolution depth images and can be efficiently used in an initialization scheme. The pose estimation is done in a decoupled way, i.e., first the rotation is extracted from the normals and then the translation is gathered from the normals and depth. A simplified scheme of the pose estimation pipeline is given in figs. 4.1 and 4.2. The only assumed hypothesis, in the case of general scenes, is that the frames contain piece-wise planar regions with co-visibility (overlapping). The term "overlapped" means a surface with co-visibility between the two points of view. We will implicitly suppose that the geometry of the scene can be approximated by piece-wise planar patches, since the surface pattern with minimal parametrization are planes.

The remainder is organized as follows. First, we discuss some related works in section 4.1.1. A review of some elementary properties is given in section 4.1.3 followed by the normal computation in section 4.1.3.1. In the sequence, we introduce the rotation estimation in section 4.2.1.2. The translation is subsequently described in section 4.2.2. The limitations and observability conditions of the approach are discussed in sections 4.2.1.4, 4.2.2.1 and 4.3. Experimental results are presented in section 4.4 for simulated and real indoor environments

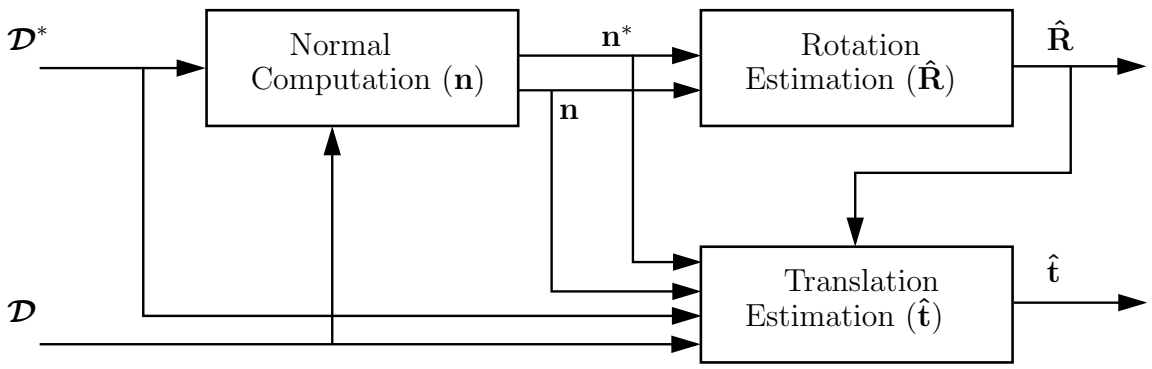

Figure 4.1 – Pose computation pipeline stages. The pose is computed in a decoupled way using the distributions of the normals, which is very efficient and can be used inside an initialization framework to direct methods.

using fisheye and spherical images and we conclude the chapter in section 4.5.

4.1.1 Related Works

The pose estimation described in this chapter is closely related to the methods of [Stoyanov et al., 2012, Ma et al., 2016] for point-cloud registration, [Fernandez-Moral et al., 2014] for automatic LIDAR/RGB-D non-overlapping camera calibration and [Zhou et al., 2016a] for rotation tracking in piece-wise planar environments. The approach in [Stoyanov et al., 2012] explored the Three-Dimensional Normal-Distributions Transform (3D-NDT) to describe the scene using distributions of geometric features (corners, planes, lines) for loop closure. [Ma et al., 2016] proposed a decoupled rotation and translation estimation using the normals from two point clouds. The rotation estimation tracks the peak of the normal distributions using a decomposition similar to the one presented in this work. However, their rotation estimation implicitly assumes a predominant rotation only in the Z axis. Furthermore, this technique is only valid to small translations and it does not explore multi-resolution for efficiency. Similarly, [Zhou et al., 2016a] estimate the rotation (no translation) from a set of dominant planes in the scene. Their algorithm starts by extracting the principal orientations of the normals of the environment and tracks the modes over time. The association, between the normals belonging to the modes in the current and in the reference frames, is done by considering the closest mode with a geodesic distance in the unit sphere. Once the association is performed, the rotation estimation is based on the same formulation presented in [Fernandez-Moral et al., 2014]. In [Fernandez-Moral et al., 2014], a rough guess of the relative rotation between the current and reference frames is provided by the user for calibrating non-overlapping RGB-D cameras. Once the association of co-planar regions is established, an elegant modified version of Arun's algorithm of ICP point-to-point [Arun et al., 1987] is derived to find the rotation in a least square sense. Here instead, we proceed with a different strategy and formulation. First, we do not insert a rough guess or assume infinitesimal/small changes in the rotation (remind that a rough relative motion is what we seek). Besides that, conversely to [Ma et al., 2016, Zhou et al., 2016a], we also derive a closed form for the translation and analyze the limitations and what is the expected performance of the approach in a set of scene configurations. Some other interesting works assume further hypothesis in the scene geometry, as the Manhattan World assumption in [Zhou et al., 2016b] for depth registration using principal component analysis of the normal vectors.

This work is also related to global registration techniques as [Li and Hartley, 2007, Yang et al., 2016] for 3D registration or [Scaramuzza and Siegwart, 2008, Berenguer et al., 2015] using appearance-based global descriptors. [Scaramuzza and Siegwart, 2008, Berenguer et al., 2015] used the appearance of omnidirectional images as a visual compass. In [Berenguer et al., 2015], a 2D rotation is estimated using a Fourier transform of the Radon descriptors for each image. From the correlation of the two Fourier transforms, a 2D rotation is computed as

the phase shift of the signals. Subsequently, the maximum correlation is used as an heuristic metric to measure the translation of the camera. Similarly, [Scaramuzza and Siegwart, 2008] performed a unidimensional correlation of an image template along the omnidirectional target image. This rough rotation estimate is used as initialization of a direct image registration. Conversely to these methods, our approach does not assume 2D motions for computing the rough pose between the frames. In [Li and Hartley, 2007], a global optimization framework is proposed for recovering the relative pose from 3D point clouds. The optimization is performed using a Bound-and-Branch paradigm with an extensive search in $\mathbb{SO}(3)$ for the rotation using an octree data structure. A Lipschitz mathematical formulation is then applied to reduce the search space in the octree. The error for each candidate rotation is computed using an ICP point-to-point where the 3D point correspondences are done by the Hungarian algorithm. Besides its theoretical and elegant formulation, their algorithm was only able to handle 3D points without any occlusions or outliers and with a running time that spans in the order of minutes for a hundred of 3D points. This makes this algorithm unrealistic as a plausible candidate for an initialization technique. An extension of their work was proposed in [Yang et al., 2016] (Go-ICP), at this time for the full six DOF pose in $\mathbb{SE}(3)$ but with the same computation drawback. In our work, a sampling of the rotation domain is also performed to compute different initialization candidates, however, our approach is much more efficient by using the normals to compute each pose candidate.

4.1.2 Contributions

The contribution of the chapter is an efficient decoupled rotation and translation estimation from the normal vectors extracted from a range image. This formulation is an alternative way of registering point clouds, which is very efficient and can be used in an initialization scheme for direct RGB-D methods. The proposed rotation estimation is related to two recent works of [Ma et al., 2016, Zhou et al., 2016a]. Conversely to these previous works, we do not assume infinitesimal motions, neither a rotation order or the Manhattan-World scene hypothesis. Furthermore, we also propose a closed form translation estimation and we analyze the observability and limitations of the method.

4.1.3 Preliminaries

This subsection summarizes some useful properties that are extensively used during this chapter. First, only the depth image is considered in the spherical frame, i.e., $\boldsymbol{\mathcal{S}}$ refers to a spherical depth image $\boldsymbol{\mathcal{D}} \in \mathbb{R}_{+}^{m \times n}$. Again, the mapping between 3D Cartesian coordinates $\mathbf{P} \in \mathbb{R}^3$ and frame pixel coordinates $\mathbf{p} \in \mathbb{P}^2$ is given by $\mathbf{P}(\mathbf{p}) = \boldsymbol{\mathcal{D}}(\mathbf{p})\Pi_S^{-1}(\mathbf{p})$, with the unit vector $\Pi_S^{-1}(\mathbf{p}) \in \mathbb{S}^2$ being the viewing direction of the 3D point $\mathbf{P}$ (see fig. 4.2 for the geometry of two 3D points viewed from the X sensor direction in different frames). If the motion between the

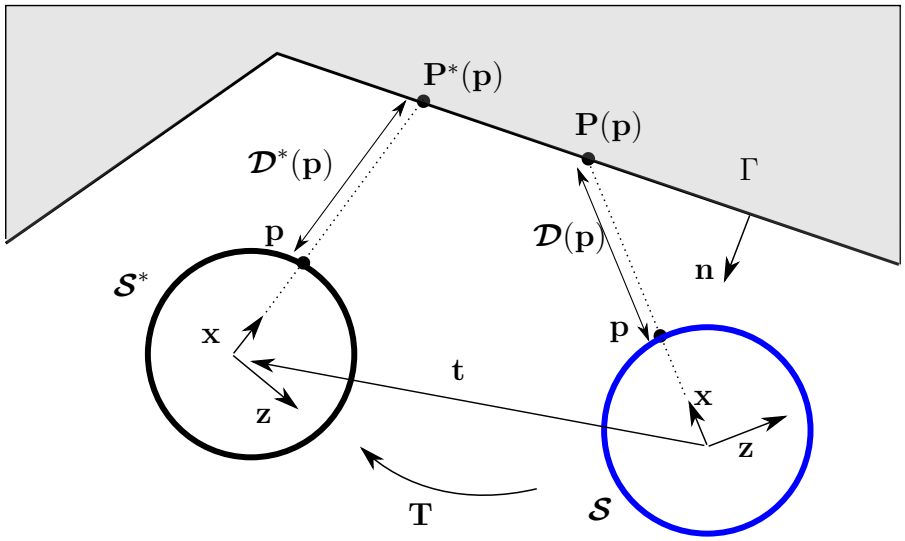

Figure 4.2 – Bird's-eye view schematic of two spherical frames $\mathcal{S}^*$ and $\mathcal{S}$ observing a planar region Γ. See the text in section 4.1.3 for notation details.

frames is known, the correspondence of image pixels between frames, under visibility conditions, is given by the spherical warping function $w : \mathbb{P}^2 \times \mathbb{R}_+ \times \mathbb{SE}(3) \mapsto \mathbb{P}^2$ as:

$$w(\mathbf{p}, \mathbf{T}) = \Pi_S \left(\| \mathbf{R} \mathcal{D}(\mathbf{p}) \Pi_S^{-1}(\mathbf{p}) + \mathbf{t} \|_S \right) \tag{4.1}$$

where Π_S is the mapping from Cartesian to pixel coordinates as in eq. (2.6) and $\mathbf{T}(\mathbf{x}) = (\mathbf{R}, \mathbf{t}) \in \mathbb{SE}(3)$ is the relative pose (rotation and translation) between the spherical frames. We introduce now the two basic geometric concepts between a rotation and two given unit vectors $\mathbf{n_1}, \mathbf{n_2} \in \mathbb{R}^3$ that will be substantially used in the next sections. The angle Θ and orthogonal axis $\mathbf{n}_\Theta$ (perpendicular to the plane formed by the two vectors) is given by:

$$\Theta = \arccos(\mathbf{n_1^T n_2}) \text{ and } \mathbf{n}_\Theta = \| \mathbf{S}(\mathbf{n_1}) \mathbf{n_2} \|_S \tag{4.2}$$

where $\mathbf{S}(\mathbf{n_1})$ represents the skew-symmetric matrix of the vector $\mathbf{n_1}$, i.e. the cross product $\mathbf{n_1} \times \mathbf{n_2} = \mathbf{S}(\mathbf{n_1})\mathbf{n_2}$. The rotation $\mathbf{R}$ thereby establishing $\mathbf{n_1} = \mathbf{R}\mathbf{n_2}$ is

$$\mathbf{R} = \exp(\mathbf{S}(\Theta \mathbf{n}_\Theta)) \tag{4.3}$$

which can be computed using the well known Rodrigues' formula (3.16). For numerical stability of (4.2), $\mathbf{R}$ is the identity matrix if $||\Theta||_1 < 0.001$ degrees. The following useful matrix properties will be also used along the chapter:

$$\begin{aligned} &\text{P1: } \mathbf{n_1} \times \mathbf{n_2} = \mathbf{S}(\mathbf{n_1})\mathbf{n_2} = -\mathbf{S}(\mathbf{n_2})\mathbf{n_1} = \mathbf{S}(\mathbf{n_2})^T \mathbf{n_1} \\ &\text{P2: } d(\mathbf{R}(\mathbf{x})\mathbf{n}) = \mathbf{R}(\mathbf{x})\mathbf{S}(\mathbf{x})\mathbf{n} = \mathbf{S}(\mathbf{x})\mathbf{R}(\mathbf{x})\mathbf{n} \\ &\text{P3: } \mathbf{S}(\mathbf{n_1})\mathbf{S}(\mathbf{n_2}) = \mathbf{n_1}\mathbf{n_2}^T - \mathbf{n_1}^T\mathbf{n_2}\mathbf{I}_{(3\times3)}. \end{aligned} \tag{4.4}$$

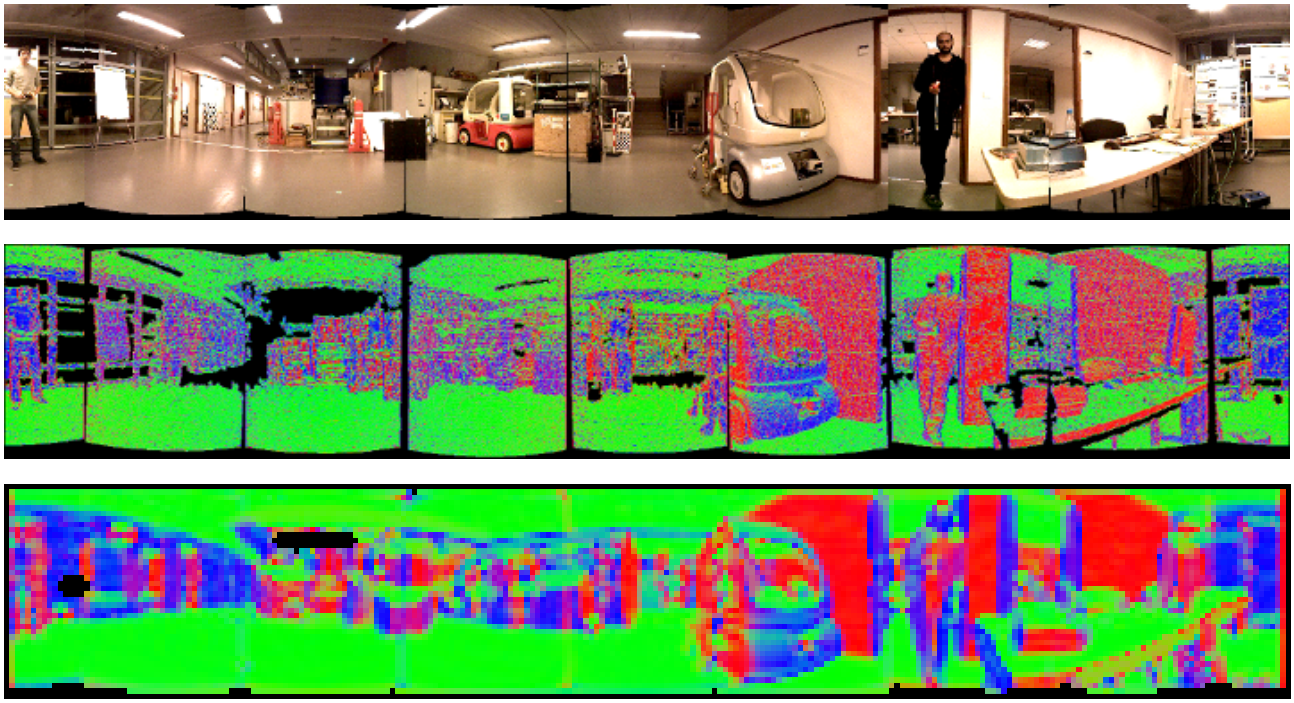

Figure 4.3 – Normal vector estimation of the raw depth in the highest resolution (middle) and downsampled depth (bottom) of an indoor scene. The colors encode the normal orientation of the surface. The improvement of the down-sampling is specially noticed by the more homogeneous colors inside planar regions. Nevertheless, the downsampling also induces border artifacts at object edges.

By last, the superscript $*$ designates variables in the reference frame $\boldsymbol{\mathcal{S}}^*$.

4.1.3.1 Normal Surface Estimation

The normals are computed using the central gradient as discussed in section 2.5. This normal computation is very efficient since we deal with ordered depth images. The pose estimation also admits low resolution images. The advantages of using low resolution depth images are twofold. First, it maintains the efficiency of the algorithm, since a reduced number of operations are performed. Note that, multi-resolution images are usually computed for direct methods, independently from the initialization. Second, the simple local derivative (the central gradient) in planar surfaces is more robust to noise in the down-sampled depth. We employed a Gaussian pyramid of four level, where each level reduces its input to a quarter of its resolution. The lowest resolution depth images were used in the results section. An example of the resulting normals in the highest and lowest resolutions are shown in fig. 4.3. This algorithm of normal vector computation works well in our experiments, still other normal computation algorithms could also be explored, as for instance, the ones presented in [Badino et al., 2011b, Klasing et al., 2009].

4.2 Decoupled Pose Estimation from Normals

Before introducing the initialization scheme to direct methods, we describe how to efficiently estimate a rough approximation of the pose using the normal vectors of two depth images. The pose estimation comprises two main sequential stages: one for the rotation and one for the translation. The unique assumed hypothesis is that the scene contains co-visible planar regions. This assumption is discussed in more detail in section 4.3. We start presenting two possible techniques for the rotation estimation in the following sections.

4.2.1 Rotation Estimation

This section presents the framework and some discussion of the use of normal vectors for the rotation estimation. The relationship of the rotation $\mathbf{R}(\boldsymbol{\omega})$ and the angle between two corresponding normals in the reference and current frames is:

$$\mathbf{n}(w(\mathbf{p}, \mathbf{R}(\boldsymbol{\omega})))^T \mathbf{R}(\boldsymbol{\omega}) \mathbf{n}^*(\mathbf{p}) = \cos(\Theta) = (\mathrm{tr}(\mathbf{R}(\boldsymbol{\omega})) - 1)/2. \tag{4.5}$$

The following two plausible normal error metrics are considered:

$$\mathbf{e}_{N1}(\mathbf{p}, \boldsymbol{\omega}) = \mathbf{n}(w(\mathbf{p}, \mathbf{R}(\boldsymbol{\omega}))) - \mathbf{R}(\boldsymbol{\omega}) \mathbf{n}^*(\mathbf{p}) \tag{4.6}$$

$$\mathbf{e}_{N2}(\mathbf{p}, \boldsymbol{\omega}) = \mathbf{n}(w(\mathbf{p}, \mathbf{R}(\boldsymbol{\omega}))) \times \mathbf{R}(\boldsymbol{\omega}) \mathbf{n}^*(\mathbf{p}) \tag{4.7}$$

A first natural question concerns the choice of the most appropriate error and their relationship with the angle in eq. (4.5). The criteria of selection of the error can be based on the convergence domain and the minimum distinguishability for each error. Let's assume corresponding normals of a single plane and then using property P1 and P3 from (4.4) and developing:

$$\begin{aligned} \frac{1}{2} \mathbf{e}_{N1}^T \mathbf{e}_{N1} &= 1 - \mathbf{n}^T \mathbf{R}(\boldsymbol{\omega}) \mathbf{n}^* = 1 - \cos(\Theta) \\ \mathbf{e}_{N2}^T \mathbf{e}_{N2} &= 1 - \mathbf{n}^{*T} \mathbf{R}(\boldsymbol{\omega})^T \mathbf{n} \mathbf{n}^T \mathbf{R}(\boldsymbol{\omega}) \mathbf{n}^* = 1 - \cos(\Theta)^2 = (1 + \cos(\Theta))(1 - \cos(\Theta)) \end{aligned} \tag{4.8}$$

that is

$$||\mathbf{e}_{N2}||_2^2 = \frac{(1 + \cos(\Theta))}{2} ||\mathbf{e}_{N1}||_2^2. \tag{4.9}$$

In other words, $||\mathbf{e}_{N1}||_2$ is an upper bound of $||\mathbf{e}_{N2}||_2$. An example of the convergence domain for a real scene containing multiple planes is presented in fig. 4.4, where each row corresponds to one cost function. These results clearly identify that the error terms (4.6) and (4.7) have similar convergence domain. We shall observe some "flatness" in both considered costs. Although this

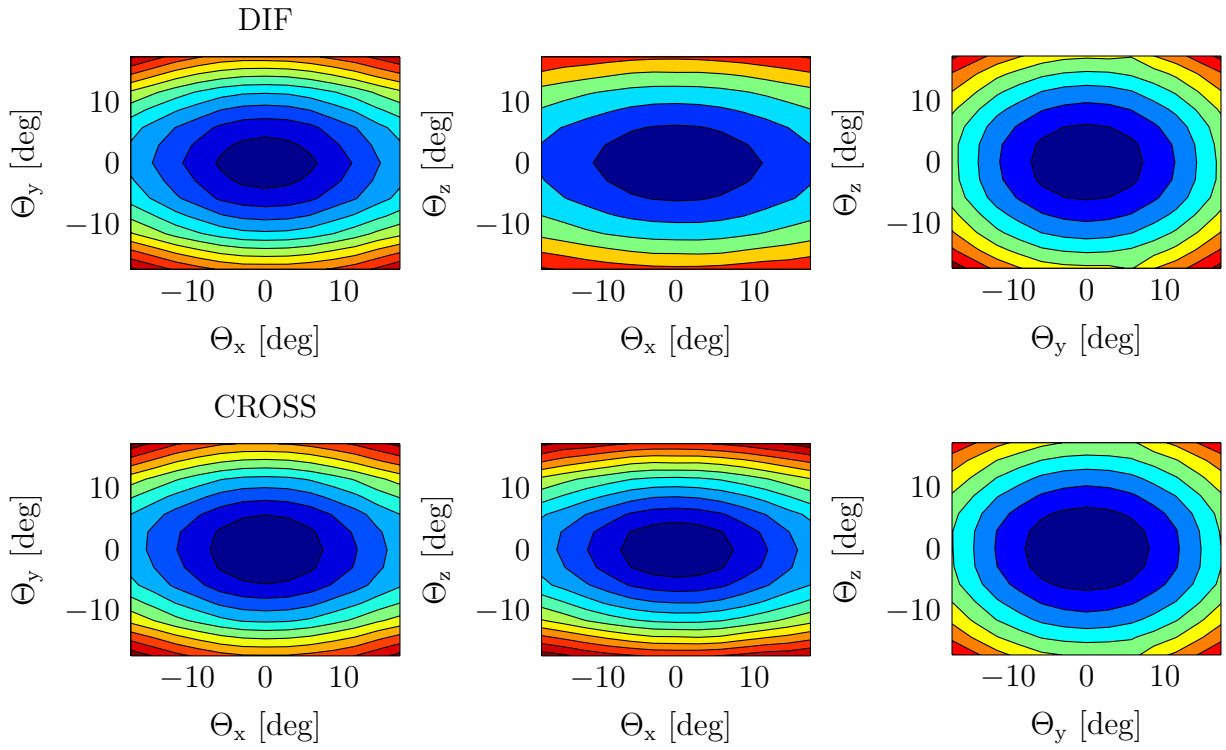

Figure 4.4 – Convergence regions for the two cost functions using only pixels with overlapped surfaces. Each row is the corresponding level curves of the costs (4.6) and (4.7) evaluated in slices of $\mathbb{SO}(3)$. The computation of the level curves is within a window neighbourhood width of 20 degrees around the solution.

"flatness" effect is not of big concern in ideal cases, it needs to be considered in the presence of noise.

4.2.1.1 Case 1: Rotation from Gradient-Based Minimization

Similarly to the direct image registration framework described in chapter 3, the cost using the normals can be seen as:

$$C = \min_{\boldsymbol{\omega}} \sum_{\mathbf{p}} \rho(\mathbf{e}(\mathbf{p}, \boldsymbol{\omega})) \tag{4.10}$$

and errors of the normals presented in eqs. (4.6) and (4.7) can be modeled as:

$$\mathbf{e}_{N1}(\mathbf{p}, \boldsymbol{\omega}) = \mathbf{n}(w(\mathbf{p}, \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))) - \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}) \tag{4.11}$$

$$\mathbf{e}_{N2}(\mathbf{p}, \boldsymbol{\omega}) = \mathbf{n}(w(\mathbf{p}, \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))) \times \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}) \tag{4.12}$$

where $\hat{\mathbf{R}}$ is a first rotation approximation. Each cost function, using $\mathbf{e}_{N1}$ and $\mathbf{e}_{N2}$, are minimized with a Gauss-Newton formulation. Therefore, we need to compute the first order Taylor expansion of the errors, i.e., the Jacobians as described in section A.3 of appendix A. In general, a few iterations were needed to reach convergence (less than 10) as depicted in the registration

example shown in fig. 4.4. The stability and convergence speed were similar for both normal errors $\mathbf{e}_{N1}$ and $\mathbf{e}_{N2}$. This minimization, using direct registration, is however computationally expensive since, for a number N of pixels, the Jacobians have $9N$ elements for both (4.6) and (4.7). In the next sections of this chapter, a more efficient formulation is exploited for estimating the rotation using the normals distributions, instead of using this iterative and local linearized costs.

4.2.1.2 Case 2: Rotation Estimation from Distributions

In the previous section 4.2.1.1, we described how to compute the rotation from the normals using an iterative gradient-based registration, similarly to the registration presented in chapter 3. In this section, we describe the rotation estimation from a more geometric point-of-view, using the concept of distributions. We start describing the rotation estimation for general scenes, i.e., without the assumption of dominant directions in the normals. In presence of planar surface regions with co-visibility/overlapped (see fig. 4.2), the following holds: $\mathbf{n}(\mathbf{p}) = \mathbf{R}\mathbf{n}^*(\mathbf{p})$. The hypothesis of overlapped planes is quite realistic since most scenes have planar surfaces (the limitations of this hypothesis will be discussed later in section 4.3). Given the normals, the product $\arccos(\mathbf{n}^*(\mathbf{p})^T\mathbf{n}(\mathbf{p}))$ is the rotation angle around the axis $\mathbf{n}^*(\mathbf{p}) \times \mathbf{n}(\mathbf{p})$. Thus, a same overlapped pixel have different possible rotations matrices, depending on the axis of rotation. Furthermore, a vector is invariant to a rotation around a parallel axis. Hence, an intermediary representation is used, for instance a decomposition, to find the overlapped regions.

Since any rotation can be decomposed as three instantaneous rotations around three orthogonal axes, from Euler's rotation theorem, we will perform projections of all normals in three subspaces to identify the planar overlapped regions, which are rotated by the same angle in this intermediary representation. For simplicity, we select the coordinate system of the current frame $\boldsymbol{\mathcal{S}}$ to define the projection operator around each axis as:

$$\text{proj}_x(\mathbf{n}) = \|(\mathbf{0}\ \mathbf{e_y}\ \mathbf{e_z})^T\mathbf{n}\|_S;\ \text{proj}_y(\mathbf{n}) = \|(\mathbf{e_x}\ \mathbf{0}\ \mathbf{e_z})^T\mathbf{n}\|_S;\ \text{proj}_z(\mathbf{n}) = \|(\mathbf{e_x}\ \mathbf{e_y}\ \mathbf{0})^T\mathbf{n}\|_S \quad (4.13)$$

with $\mathbf{e_x} = (1\ 0\ 0)^T$, $\mathbf{e_y} = (0\ 1\ 0)^T$, $\mathbf{e_z} = (0\ 0\ 1)^T$ and $\mathbf{0} = (0\ 0\ 0)^T$. The corresponding instantaneous rotation angle of each projection $\omega_x,\ \omega_y,\ \omega_z \in [0, \pi)$ is given by the scalar products:

$$\begin{aligned}
\omega_x(\mathbf{p}) &= \arccos(\text{proj}_x(\mathbf{n}^*(\mathbf{p}))^T\text{proj}_x(\mathbf{n}(\mathbf{p}))) \\
\omega_y(\mathbf{p}) &= \arccos(\text{proj}_y(\mathbf{n}^*(\mathbf{p}))^T\text{proj}_y(\mathbf{n}(\mathbf{p}))) \\
\omega_z(\mathbf{p}) &= \arccos(\text{proj}_z(\mathbf{n}^*(\mathbf{p}))^T\text{proj}_z(\mathbf{n}(\mathbf{p})))
\end{aligned} \quad (4.14)$$

In the same way, the sign of each rotated angle obeys the sign of the projections cross product:

$$s_i = \text{sign}(\mathbf{e}_i^T \text{proj}_i(\mathbf{n}^*(\mathbf{p})) \times \text{proj}_i(\mathbf{n}(\mathbf{p}))) \tag{4.15}$$

for $i = \{x, y, z\}$ as for the angles in eq. (4.14). Assuming that the scene contains overlapping planar regions, an estimation of the rotation can be obtained from the projection angles of all pixels using (4.13), (4.14) and (4.15). These angles can be seen as three distributions and the property we explore to extract the points with co-visibility is that overlapped planes of a same surface are rotated by the same projected angles. For instance, this is performed by finding the sub-set of pixels $\mathbf{p}^+$ belonging to the peaks of the three distributions simultaneously, i.e., the modes in the distributions in fig 4.5. With the sub-set of pixels $\mathbf{p}^+$ (inliers points), one can find the median angle of each projection:

$$\hat{\omega}_i = \text{median}(s_i(\mathbf{p}^+)\omega_i(\mathbf{p}^+)) \tag{4.16}$$

with $i = \{x, y, z\}$. Then, the rotation in the axis/angle form is given by $\boldsymbol{\omega} = (\hat{\omega}_x \ \hat{\omega}_y \ \hat{\omega}_z)^T$ and the equivalent rotation matrix is recovered by the exponential mapping $\hat{\mathbf{R}} = \exp(\mathbf{S}(\boldsymbol{\omega}))$. This algorithm is much more efficient and accurate than the gradient-based optimization described in section 4.2.1.

Rotation from the Matching of Distributions

An interesting particular scenario is of scenes with normals in dominant directions, and with predominantly rotations in one direction, as discussed in [Ma et al., 2016]. We introduce and give further details of this algorithm because it establishes a different point-of-view of the use of distributions to find corresponding pixels between the depth images. In this case, one can try to match the surface regions, by computing six distributions of the normals at the reference and current frames. Since there might not be overlapping between the surfaces, the decomposition needs to be performed in a slightly different manner than in the case of general scenes. The surfaces orientations are used as a compass, similar to 1D correlation on images, where the three distributions for the current frame are:

$$\begin{aligned}
\omega_x(\mathbf{p}) &= \arccos(\mathbf{e}_y^T \text{proj}_x(\mathbf{n}(\mathbf{p}))), \quad s_x = \text{sign}(\mathbf{e}_x^T \mathbf{e}_y \times \text{proj}_x(\mathbf{n}(\mathbf{p}))) \\
\omega_y(\mathbf{p}) &= \arccos(\mathbf{e}_z^T \text{proj}_y(\mathbf{n}(\mathbf{p}))), \quad s_y = \text{sign}(\mathbf{e}_y^T \mathbf{e}_z \times \text{proj}_y(\mathbf{n}(\mathbf{p}))) \\
\omega_z(\mathbf{p}) &= \arccos(\mathbf{e}_x^T \text{proj}_z(\mathbf{n}(\mathbf{p}))), \quad s_z = \text{sign}(\mathbf{e}_z^T \mathbf{e}_x \times \text{proj}_z(\mathbf{n}(\mathbf{p})))
\end{aligned} \tag{4.17}$$

The remaining three distributions in the reference frame are computed by just replacing $\mathbf{n}(\mathbf{p})$ by $\mathbf{n}^*(\mathbf{p})$ in (4.17). Subsequently, we select the rotation axis with unimodal distribution and

then compute the rotation angle:

$$\hat{\omega}_i = \text{median}(s_i \omega_i) \text{ and } \mathbf{R}_i = \exp(\mathbf{S}(\mathbf{e}_i \hat{\omega}_i) \tag{4.18}$$

where i is the selected axis. After updating the reference normals $\mathbf{n}_w^* = \mathbf{R}_i \mathbf{n}^*$, the reference distributions are recomputed using eq. (4.17). This process is repeated for the two remaining axes. Assuming that the computation order is Y, Z and then X, leads to a rotation: $\mathbf{R} = \mathbf{R}_x \mathbf{R}_z \mathbf{R}_y$. Observe that this is similar to a 1D template-based convolution in intensity images [Scaramuzza and Siegwart, 2008].

Under the presence of three distinct modes, panoramic images and small translations, this case allows estimating any rotation, even for frames without surface overlapping.

4.2.1.3 Which Rotation Estimation Should Be Used?

A natural question is which rotation formulation should be preferred in terms of convergence domain, efficiency and accuracy. As discussed in section 4.2.1.1 the gradient-based optimization is not efficient and assumes small rotations to converge. We will introduce the advantages and drawbacks between the overlapping and mode tracking described in section 4.2.1.2 with two representative examples. The first example is composed of frames with a rotation $\boldsymbol{\omega} = (-16 \ 26 \ -4)^T$ degrees, as shown in fig. 4.5. The first row depicts the reference spherical frame and the second row the current frame. The corresponding normals are encoded by color in the second column. We show the respective RGB images, in the first column, only for visualization purposes since they are not used in the computation. The last row depicts the distributions of each projected angle using the overlapped normals. This results in an estimated rotation of around $\boldsymbol{\omega} = (-17 \ 24 \ -2)^T$ degrees. This example depicts a successful estimation in the three axes. The six distributions for the mode tracking are depicted in fig. 4.6. The rotation estimation using mode tracking, however, cannot select the corresponding mode in Y and therefore it converges to $\boldsymbol{\omega} = (39 \ -42 \ 36)^T$ degrees. This is mainly due to the translation between the frames, of $\mathbf{t} = (-1.6 \ 0.1 \ 0.3)^T$ meters, which reshapes the modes.

To exemplify the convergence domain of the methods, we compute the rotation after "virtually rotating" the reference frame for different rotations Θ around the Y axis ($-90 < \Theta < 90$). We compute the error for each method as:

$$e_R = \arccos(\text{tr}(\bar{\mathbf{R}}^T \mathbf{R}(\boldsymbol{\omega})) - 1)/2) \tag{4.19}$$

where $\bar{\mathbf{R}}$ is the real rotation between the frames. The angle errors of each method can be seen in fig. 4.7, where the overlapping corresponds to the blue curve and the mode [Ma et al., 2016] is in green. Due to the presence of multiple modes in Y, the mode tracking does not estimate the real rotation for all angles. The domain of convergence around the Y for this scene using

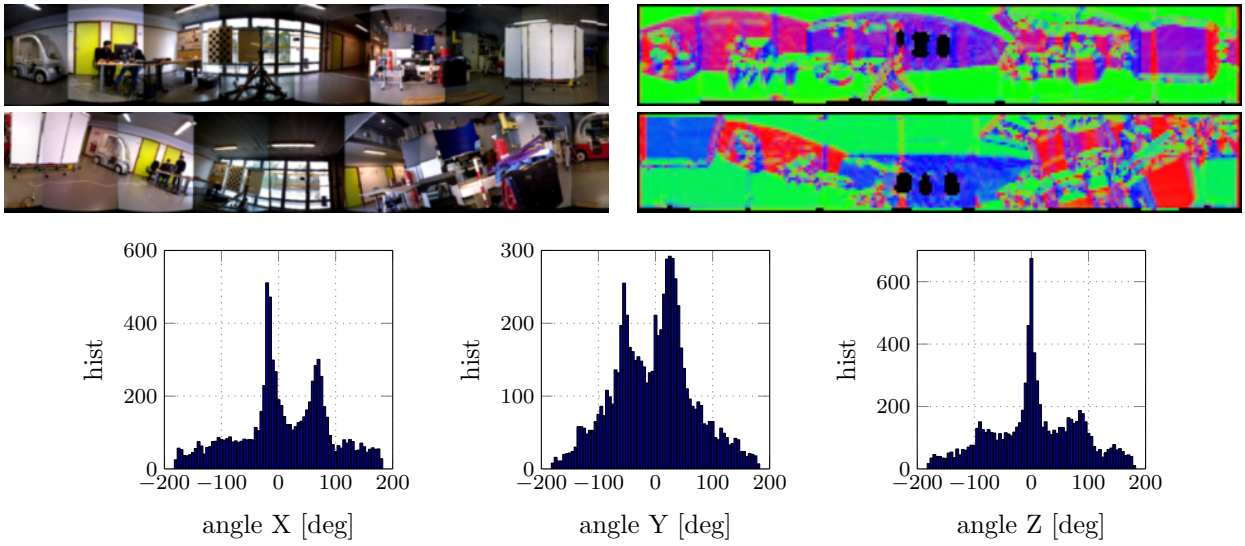

Figure 4.5 – Rotation estimation example for two real frames with rotation $\boldsymbol{\omega} = (-16\ 26\ -4)^T$ using the overlapping. The first row depicts the reference and the second row the current frames. The corresponding normals are encoded by color in the second column. The last row depicts the distributions for each projected angle using the overlapping of the normals. See the text of section 4.2.1.3 for details.

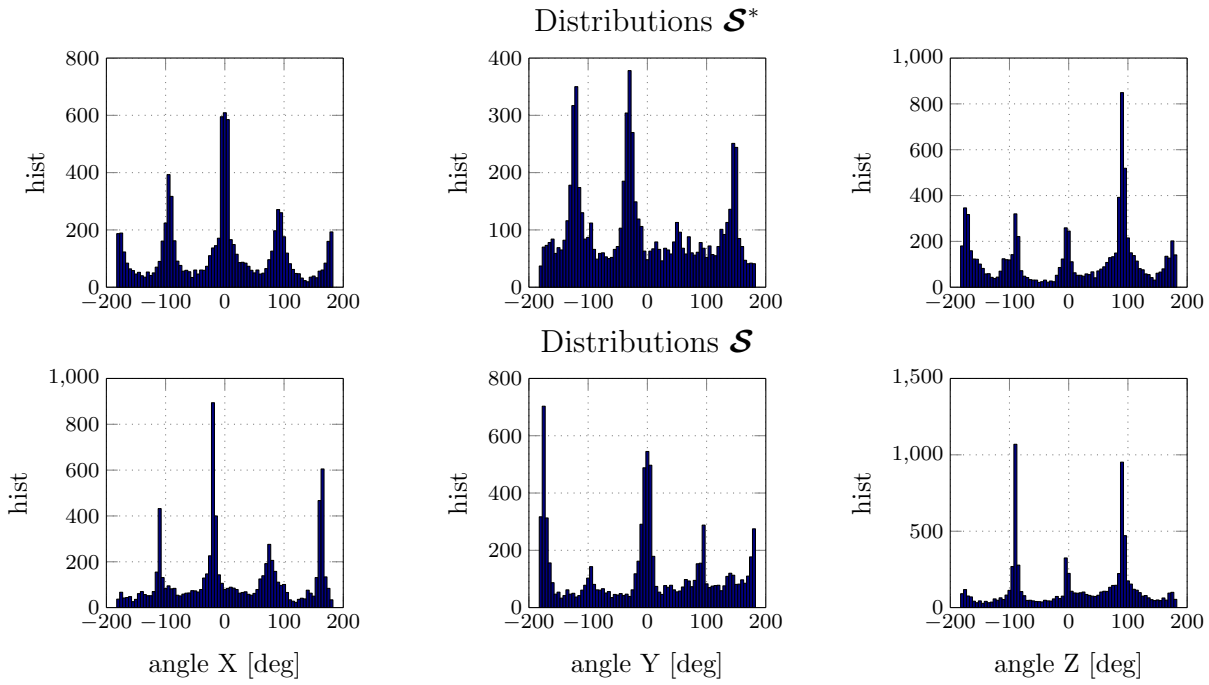

Figure 4.6 – Rotation estimation example using the mode distribution tracking. The first row corresponds to the reference and the second row to the current frame distributions. See the text of section 4.2.1.3 for details.

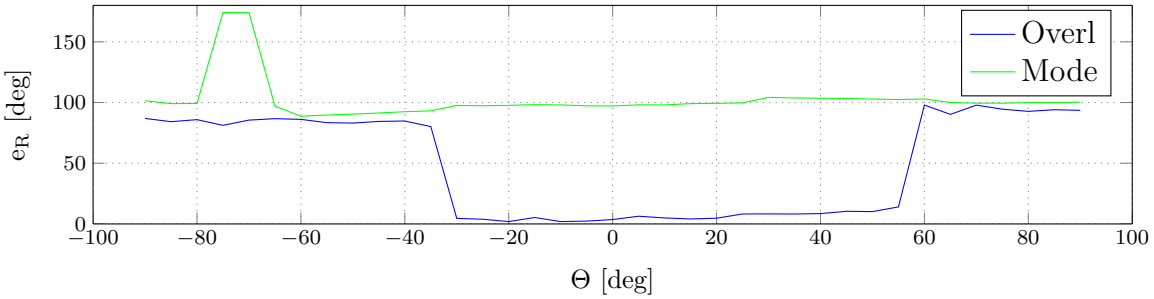

Figure 4.7 – Convergence domain for the registration of the frames depicted in fig. 4.5. The green curve shows the angle error of the 3D rotation using the mode tracking, which was not capable of estimating rotation. The overlapping is depicted in blue and has a convergence domain of around 90 degrees.

the angle overlapping is almost 90 degrees. Observe that the domain of convergence is not symmetric in this scene, such domain depends on the particular pair of depth images being registered.

The second example is composed of frames with a rotation $\boldsymbol{\omega} = (0\ -180\ 0)^T$, as shown in fig. 4.8. The first row depicts the reference and the second row the current frames. The third row depicts the distributions of each projected angle using the overlapped normals. This results in an estimated rotation of around $\boldsymbol{\omega} = (-1\ -1.5\ 1)^T$ degrees, because the overlapping property is not fulfilled for this initial configuration. The rotation estimation using mode tracking can select the right modes and hence it converges to $\boldsymbol{\omega} = (-0.5\ -177\ -0.5)^T$ degrees. In the same way than for the previous example, we display the convergence domain of the methods after "virtually rotating" the reference frame of Θ around the Y axis ($-180 < \Theta < 180$). The convergence domain of each method can be seen in fig. 4.10. Conversely, to the previous example, the rotation could be estimated for any angle using the mode tracking. The convergence domain using the overlapping property was of around 100 degrees. Therefore, the mode tracking is well suited with unimodal distributions and considering a small linear speed of the camera, because the translation can modify the shape of the distributions. These conditions, however, cannot be often verified in real scenes as shown in the trajectories of fig. 4.11 for the rotation using the overlapping (in blue) and the mode tracking (in green). Observe that the overlapping is clearly more accurate than the mode tracking, being capable of estimating correctly the rotation with large translations and scenes with multimodal normals. Since we aim to keep the method general, without assumptions in the motion or in the number of modes in the scene, we chose the overlapping property for the rotation estimation in the following sections.

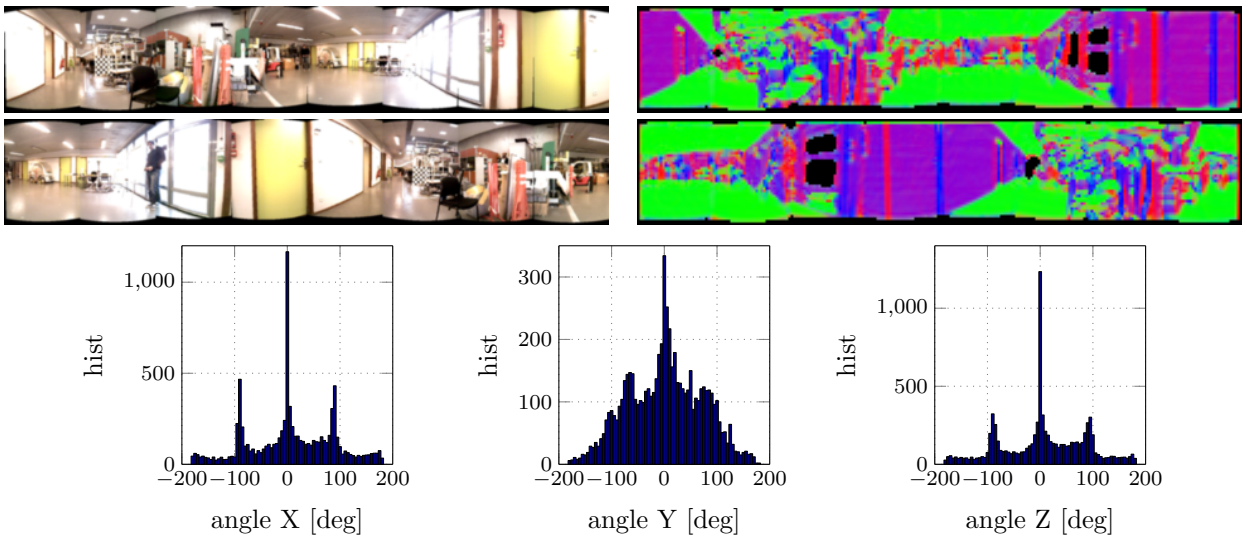

Figure 4.8 – Second rotation estimation example for two real frames with rotation $\boldsymbol{\omega} = (0 \ -180 \ 0)^T$. The first row depicts the reference and the second row the current frame. The corresponding normals are encoded by color in the second column. The last row depicts the distributions for each projected angle using the normals overlapping. See the text of section 4.3 for details.

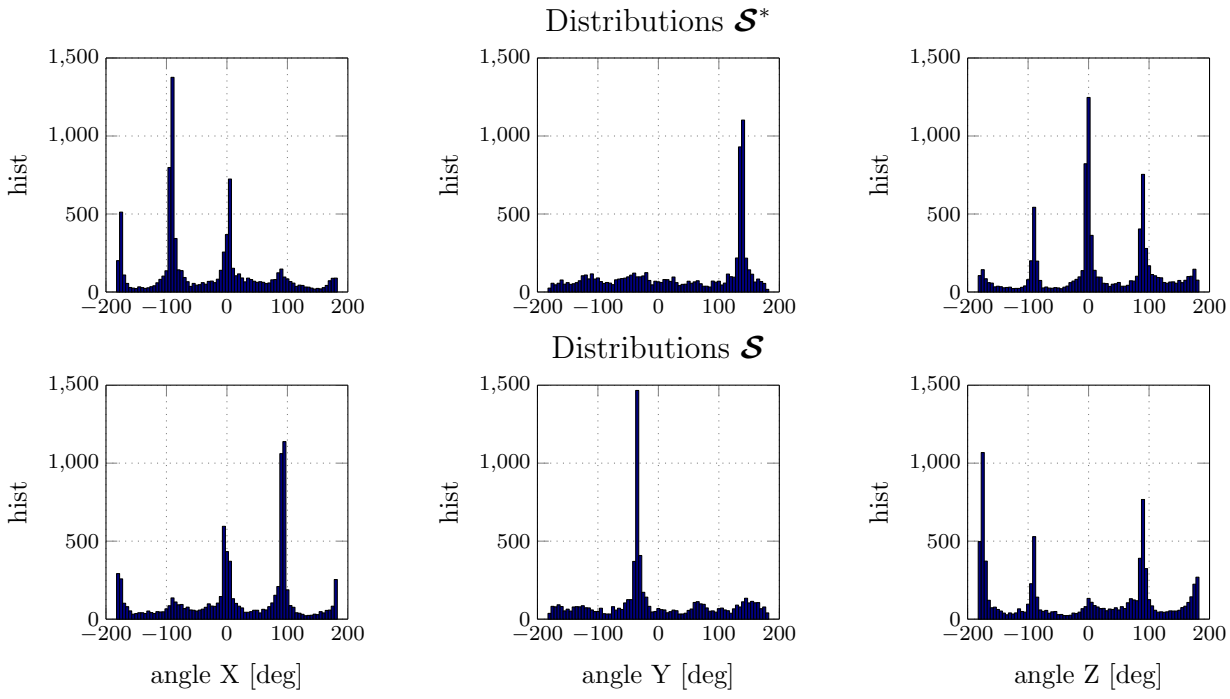

Figure 4.9 – Rotation estimation example using the mode distribution tracking. The first row corresponds to the reference and the second row to the current frame distributions. Observe that due to the geometry configuration of the scene, the distributions in Y have a well defined mode, which corresponds to the same surfaces in both frames (the purple pixels). See the text of section 4.3 for details.

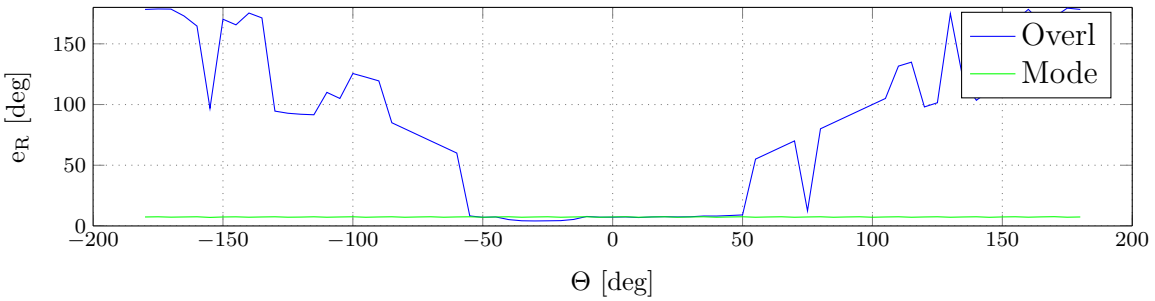

Figure 4.10 – Convergence domain for the registration of the frames depicted in fig. 4.8. The green curve shows the angle error of the 3D rotation using the mode tracking. The overlapping is depicted in blue and has a convergence domain of around 100 degrees, while the 2D mode tracking converges for any rotation, due to the small translation between the frames and because the frames have a well defined peak in the Y projections.

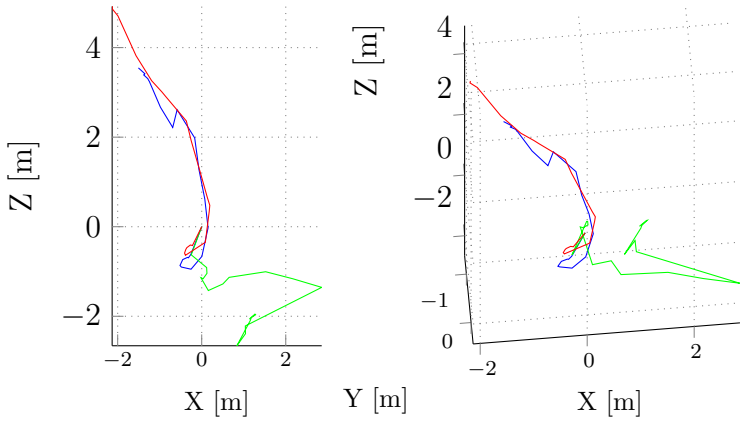

Figure 4.11 – Resulting trajectories from the pose estimation using the overlapping and the mode tracking. The ground truth trajectory is depicted in red, the green curve is using the mode tracking and the blue curve is using the overlapping. The translation changed the shape of the modes, not allowing the mode tracking to converge. The estimation using the overlapping approach is much less affected and therefore have a more accurate estimate of the motion as shown by the blue curve.

4.2.1.4 Rotation Observability

Let's suppose firstly that overlapped regions are given, i.e., starting from a set of inlier pixels $\mathbf{p}^+$. We want to find the rotation that minimizes the cost (for simplicity using the ℓ^2 norm):

$$\min_{\boldsymbol{\omega}} \sum_{\mathbf{p}\in\mathbf{p}^+} \frac{1}{2}||\mathbf{e}_{N1}(\mathbf{p},\boldsymbol{\omega})||_2^2 \tag{4.20}$$

where $\mathbf{e}_{N1}(\mathbf{p},\boldsymbol{\omega})$ is the difference error between corresponding normal vectors (4.6) observed from two distinct frames (as discussed in sections 4.2.1 and 4.2.1.2). Developing the cost (4.20) as in [Fernandez-Moral et al., 2014, Zhou et al., 2016a], the rotation is observable if the scene has, at least, three planes in linearly independent directions. This condition is often fulfilled in indoor scenarios with walls, floor and ceiling not in the same direction. Therefore, the

observability remains mainly on how to find corresponding pixels. The observability index concerning the rotation will be then related to the number of "similar" modes in the projected angle distributions.

4.2.2 Translation Initialization

We address the translation estimation in this section. After applying the rotation estimate to the reference frame (also as known as "derotation" process [Corke and Mahony, 2009]), the updated overlapped surfaces for the translation is done by checking the angle between the normals. At this time, the set of overlapped pixels $\mathbf{p}^+$ are the pixels in both frames with similar normals ($\mathbf{n}(\mathbf{p}) \approx \mathbf{n}^*(\mathbf{p})$), i.e.,

$$\mathbf{p} \in \mathbf{p}^+ \text{ if } ||\arccos(\mathbf{n}^{*T}(\mathbf{p})\mathbf{n}(\mathbf{p}))||_1 < \varepsilon_1. \tag{4.21}$$

where ε_1 is the maximum allowed angle between the normals. Hence the pixels considered to be overlapped follows the same plane equation Γ. The plane equation for the point in the pixel $\mathbf{p}$ of the current image is given by

$$\Gamma : \mathbf{n}^T(\mathbf{p})\mathbf{P}(\mathbf{p}) + d = 0 \Rightarrow \mathbf{n}^T(\mathbf{p}) \left(\boldsymbol{\mathcal{D}}(\mathbf{p})\Pi_S^{-1}(\mathbf{p})\right) + d = 0 \tag{4.22}$$

with $\Pi_S^{-1}(\mathbf{p})$ the viewing direction (in the unit sphere). Denoting the residual rotation $\mathbf{R}(\mathbf{p})$ for each pixel such as $\mathbf{n}^T(\mathbf{p}) = \mathbf{R}(\mathbf{p})\mathbf{n}^{*T}(\mathbf{p})$, the same plane viewed from the reference depth image in the direction $\Pi_S^{-1}(\mathbf{p})$ (as depicted in fig. 4.2) is therefore:

$$\Gamma : \mathbf{n}^T(\mathbf{p}) \left(\mathbf{R}(\mathbf{p})\boldsymbol{\mathcal{D}}^*(\mathbf{p})\Pi_S^{-1}(\mathbf{p}) + \mathbf{t}\right) + d = 0 \tag{4.23}$$

Subtracting the left side of eq. (4.22) and (4.23), the relationship between the normal vector, depth, viewing direction and the translation (for a pixel $\mathbf{p} \in \mathbf{p}^+$) is

$$\mathbf{n}^T(\mathbf{p})\mathbf{t} = \mathbf{n}^T(\mathbf{p}) \left(\Pi_S^{-1}(\mathbf{p})\boldsymbol{\mathcal{D}}(\mathbf{p}) - \mathbf{R}(\mathbf{p})\Pi_S^{-1}(\mathbf{p})\boldsymbol{\mathcal{D}}^*(\mathbf{p})\right). \tag{4.24}$$

Note that eq. (4.24) cannot be simplified since the scalar product $\mathbf{n}^T\mathbf{t} = \mathbf{n}^T(\mathbf{P} - \mathbf{R}\mathbf{P}^*)$ has $\mathbf{t} = \mathbf{P} - \mathbf{R}\mathbf{P}^*$ only when the translation is parallel to the normal of the plane Γ. For efficiency, the residual rotation in (4.24) is calculated for each pixel $\mathbf{p}$ using an approximation of eq. (4.3):

$$\mathbf{R}(\mathbf{p}) = \mathbf{I}_{(3\times3)} + \Theta(\mathbf{p})\mathbf{S}\left(\mathbf{n}_\Theta(\mathbf{p})\right) \tag{4.25}$$

where the angle is $\Theta(\mathbf{p}) = \arccos(\mathbf{n}^{*\mathbf{T}}(\mathbf{p})\mathbf{n}(\mathbf{p}))$ and the axis $\mathbf{n}_\Theta(\mathbf{p})$ is the orthonormal vector to $\mathbf{n}^*(\mathbf{p})$ and $\mathbf{n}(\mathbf{p})$ using eq. (4.2). In ideal conditions, i.e., depth and normals without noise and

perfect rotation estimate in section 4.2.1.2, the residual per pixel rotation $\mathbf{R}(\mathbf{p})$ is the identity matrix.

Some remarks can be drawn from equation (4.24): *i)* the system has a well defined solution if and only if there is three planes with linearly independent normals; *ii)* points with normals orthogonal to the motion do not contribute to the estimation ($\mathbf{n}^T\mathbf{t} = 0$ independently of $||\mathbf{t}||_2$) and; *iii)* a point with view direction orthogonal to the normal is ill-conditioned, i.e., $\mathbf{n}^T\Pi_S^{-1} \approx 0$ and consequently $||\boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{D}}^*||_1$ is unbounded. Thus, for avoiding outliers in the system (4.24) these points, whose angle between the normal and view direction is almost orthogonal, should not be considered (e.g., by only selecting points with $\arccos(\mathbf{n}^T\Pi_S^{-1}) < 70$ degrees). In the case where the system (4.24) is well-conditioned, it is efficiently solved using a robust M-estimator with, for instance, the Huber's loss function [Zhang, 1995]. Finding a conditioned system of equations is discussed in the next section.

4.2.2.1 Translation Observability and Conditioning

Let's consider the system of equations using eq. (4.24) for all pixels belonging to the set $\mathbf{p}^+$. This system have a unique solution if the matrix $\mathbf{N}$ for all pixels $\mathbf{p_n} \in \mathbf{p}^+$, $\mathbf{N} = [\mathbf{n(p_1)}\ \mathbf{n(p_2)}\ ...\ \mathbf{n(p_n)}]^T$ is of rank three, i.e., given at least three points from three different planes with linear independent orientations. Of course when noise is present in the normals, $\mathbf{N}$ has almost surely rank three, but then the solution of eq. (4.24) is merely an artifact produced by the noise.

Our goal is to reduce the conditioning of the matrix $\mathbf{N}$ (ratio of the maximum and minimum eigenvalues). We will proceed, in a first moment, following the works of [Meilland et al., 2015, Gelfand et al., 2003] to select the 50% salient measurements of $\mathbf{N}$ that best constraints each DOF of the system. This is done by ordering the lines of $\mathbf{N}$ such that the conditioning of the subset of equations is as close to one as possible. This conditioning also gives a measure of the normals distribution in the sphere. We will use the measure of the conditioning of the subset of salient lines $\mathbf{N_s}$ as an observability index. If the conditioning of $\text{cond}(\mathbf{N_s}^T\mathbf{N_s}) > e_2$, the system in (4.24) is said to have a "ill-conditioned geometry" and we proceed to a dimension reduction. A Gaussian-Jordan elimination with partial pivoting is then used to find the column space of $\mathbf{N_s}$ and the translation estimation is done using the robust M-estimator for the two remaining DOF that are well conditioned.

4.3 Overlapping Assumption and Initialization Scheme

In this section, we will discuss what are the conditions to obtain a good pose estimation and the limits of our approach in the case of general scenes. It is natural that the observability of the computation depends on the scene geometry, i.e., in the size of the planes, their symmetry and their orientation. As stated in section 4.2.1.4, the rotation observability condition, that at least three planes have linearly independent normal vectors, is generally fulfilled for most scenes.

The observability then remains mainly in how to extract the overlapped regions, which depends directly on the scene symmetry. The property we explore to extract the overlapped regions (presented in section 4.2.1.2) is that planes with co-visibility are rotated by the same angle. The angles are then represented as distributions and we select the peak (the mode) as being the one corresponding to the right overlapped points. The distributions, however, can have many modes in presence of geometry symmetry and the one corresponding to the real rotation can be under-represented. Some classical examples are symmetric spaces, e.g., the sphere for any rotation $||\boldsymbol{\omega}||_2 > 0$, the cylinder with $||\boldsymbol{\omega}||_2 > 0$ around the cylinder axis or the cube with rotation modulo $\text{mod}(||\boldsymbol{\omega}||_2, \pi/2) = 0$. Other examples are described in [Gelfand et al., 2003]. In fact, any rotation bigger than $||\boldsymbol{\omega}||_2 > \pi/2$ in the cube is not observable since the overlapping assumption is not fulfilled. Let's consider again the cubic scene example with the sensor in the center of the cube. As stated previously, any rotation such as $\text{mod}(||\boldsymbol{\omega}||_2, \pi/2) = 0$ cannot be observable. However, the assertion that any rotation $0 < ||\boldsymbol{\omega}||_2 < \pi/2$ can be observed is false. In fact, any rotation $||\boldsymbol{\omega}||_2$ bigger than $\pi/4$ in any direction cannot be estimated because the peak of the distribution is at $(||\boldsymbol{\omega}||_2 - \pi/2)\boldsymbol{\omega}/||\boldsymbol{\omega}||_2$. In these cases the distribution becomes bi-modal, where one of the modes corresponds to the real rotation. Hence this states that the rotation is not observable, in general, by our method. For scenes with symmetry around a defined axis, the maximum observable angle is half the period of the symmetry. Similarly, the observability is limited by the FOV of the sensor.

4.3.1 Pose Initialization Scheme

Although the pose estimation from normal vectors is not observable in general, it has a large convergence domain and is very efficient. This can be seen in the running time shown in fig. 4.12 compared to an ICP point-to-plane technique with the same low resolution depth images. In general, our convergence domain is of at least 45 degrees in the Y direction, as discussed in the examples of section 4.2.1.3. Due to the limited FOV in the vertical axis, the rotation observability is limited to 30 degrees in the X and Z axes when using the indoor images. In order to increase the basin of convergence, we can sample the space of rotations and compute different initialization candidates. For the indoor images, this requires the computation of the pose at nine different initialization candidates. From the nine candidate poses $(\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_9)$ and their related virtual depth images:

$$(\boldsymbol{\mathcal{D}}^*(w(\mathbf{p}, \mathbf{T}_1)),\ \boldsymbol{\mathcal{D}}^*(w(\mathbf{p}, \mathbf{T}_2)),\ \ ... \ , \boldsymbol{\mathcal{D}}^*(w(\mathbf{p}, \mathbf{T}_9)))$$

we can select the one that minimizes the average error between the current and virtual depth images. Due to the efficiency of the pose estimation using the normals, the computation of the nine candidates is still more efficient than a simple computation using the ICP point-to plane.

4.4 Results and Discussion

In this section, we evaluate the pose estimation in indoor simulated and real spherical sequences with challenging conditions, i.e., in scenes with corridor-like environments, large rotations and translations. We start presenting the parameters tuning used in all experiments and then the accuracy and observability of the method in some simulated sequences. Finally, we show direct registration experiments in indoor scenes with and without the initialization scheme.

4.4.1 Implementation and Parameter Tuning

The maximum depth values in our frames was of 15 meters and the sampling of the projected angle distributions (resolution of the histograms) was of 5 degrees to define the overlapping points (inlier pixels). In the translation estimation phase, points are considered to be overlapped if the angle between the normals (4.21) is of $\varepsilon_1 = 10$ degrees by considering equally the imprecision of the rotation estimation and of the normal vector computation. Finally, the ratio of salient pixels in 4.2.2.1 was set to 50% and the maximum accepted conditioning of the system without dimension reduction was heuristically set to $\varepsilon_2 = 10$.

4.4.2 Pose Estimation Results

We follow to evaluate the rotation and translation for a variety of scenes (offices, halls, corridors-like environments) using three different sequences of spherical images, to find how robust the estimation is in presence of large translations/rotations and the effects of depth noise and occlusions to the estimation. We start using fisheye depth images from the "room sequence" of [Zhang et al., 2016], as shown in the left image of fig. 4.13. For checking the performance of the estimation with large motions, we have sampled the sequences with different gaps: 3, 5, 10, 15 and 20 frames – e.g. a gap of 20 frames corresponds to calculate the motion

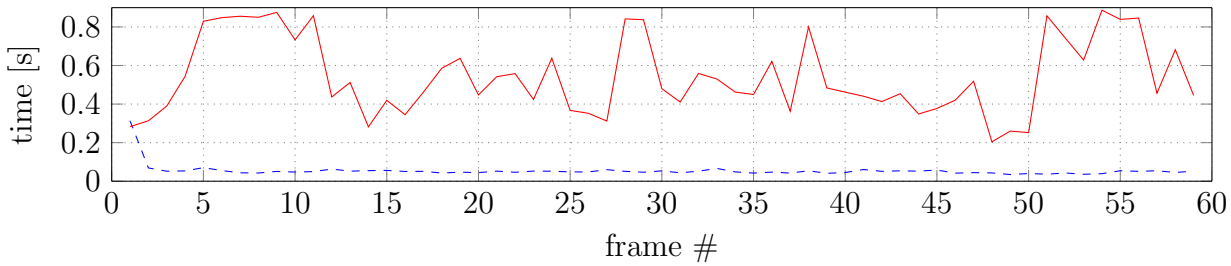

Figure 4.12 – Total running time for ICP point-to-plane (in red) and for the pose estimation from normals (in blue). The rotation estimation is done in around 0.013 seconds. The translation estimation (i.e. the conditioning and robust M-estimation with Huber) is done in 0.03 seconds. The conditions used to stop the optimization in the ICP algorithm were: maximum number of iterations of 50, norm of the rotation increment of 3 degrees and norm of the translation increment of 5 centimetres.

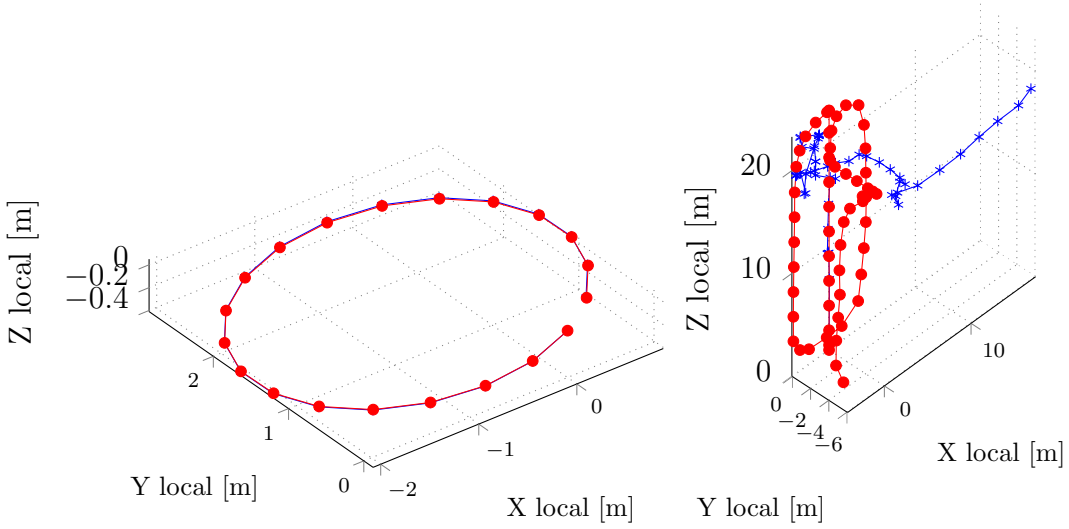

Figure 4.13 – Trajectory estimation for the simulated fisheye "room sequence" (left) and the "Atrium" spherical sequence (right), both with a gap of 10 frames. The ground truth trajectories are depicted in red (dotted) and the estimation using the normals is in blue.

Table 4.1 – Rotation and translation estimation errors for all sequences – mean absolute relative pose error (RPE), absolute standard deviation and absolute median error.

	Rot RPE [deg]			Trans RPE [m]		
	$\overline{\|\|\boldsymbol{\omega}\|\|}$	std$\|\|\boldsymbol{\omega}\|\|$	med$\|\|\boldsymbol{\omega}\|\|$	$\overline{\|\|\mathbf{t}\|\|}$	std$\|\|\mathbf{t}\|\|$	med$\|\|\mathbf{t}\|\|$
Atrium gap 5	0.25	0.39	0.06	0.11	0.18	0.04
Atrium gap 10	0.81	3.79	0.07	0.10	0.13	0.05
Atrium gap 15	3.02	12.58	0.09	0.32	0.61	0.12
Atrium gap 20	6.35	21.3	0.11	0.41	0.69	0.12
Inria1 gap 3	4.90	14.13	1.15	0.25	0.31	0.14
Inria2 gap 20	7.04	19.21	1.46	0.35	0.39	0.20

between the image pairs (1,21), (21,41), ..., $(i, i + \text{gap})$. The pose is well estimated using the fisheye sequence for all the gaps. The experiment using a gap of 10 frames is shown in fig. 4.13.

Spherical Simulated Sequence

Subsequently, we experiment with spherical depth images of the Sponza Atrium model ("Atrium" sequence), which is composed of corridors and open indoor areas. This sequence has typical symmetries that avoid the estimation of the translation component along the Z direction. The inter-frame motion in these images are of around 0.1 meters and rotation of up to 15 degrees/frame. The frame skipping results in translations up to 2.1 meters and rotations up to 70 degrees in this sequence. The relative pose errors (RPE) for these experiments are shown in table 4.1 as: the mean absolute error, the absolute standard deviation and the median absolute error. The rotation is fairly estimated in more than 99% of the cases with a gap of 10

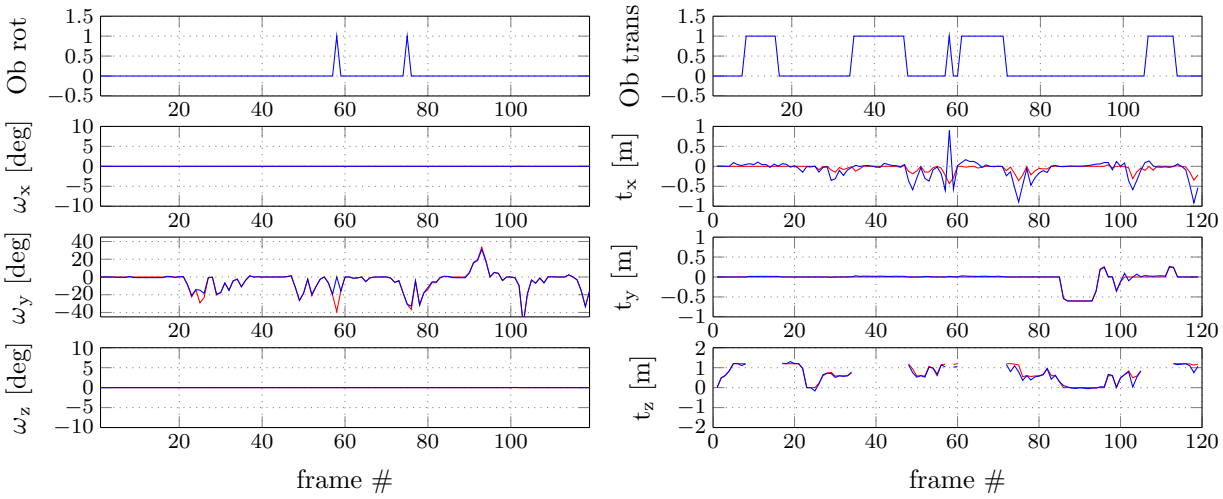

Figure 4.14 – Pose estimation results for the simulated spherical sequence with a gap of 10 frames. The ground truth poses are the curves in red and the estimation from the normals are in blue. The graphics in the first column correspond to the rotation and the second column to the translation. The first graphic at each column depicts the observability index. The rotation index is one if the distribution is bi-modal. The translation observability index is set to one, when the conditioning of the linear system for the translation is bigger than $\varepsilon_2 = 10$. See the text for details.

frames with a mean absolute error of 0.8 degrees. We show the results for the gap of 10 images in fig 4.14 and in the right image of fig. 4.13, for both rotation and translation. The method failed in 10% (6/59) of cases for the experiment with a gap of 20 frames. These cases happened when the reference frame was almost completely occluded in the current frame (e.g., 90 degrees corners) and because of the scene symmetry. These failure cases were expected to happen as discussed in section 4.3 and were detected by the observability index, which is displayed in the first plot of fig. 4.14.

The translation estimation is done after warping the reference depth image using the rotation. As stated in section 4.2.2.1, the DOF for which the FIM is ill-conditioned cannot be accurately estimated using this formulation, some examples are depicted in fig. 4.14, where the t_z component could not be estimated in the frames acquired in corridors-like scenes. These cases were also predicted in section 4.2.2.1 and the translation index show the detected cases in the first plot of the right column.

Spherical Indoor Real Sequences

We performed similar experiments using real spherical images. These real sequences were acquired in the hall and offices of the Inria building using the indoor omnidirectional RGB-D acquisition rig mounted on an holonomic mobile robot. The first real sequence (Inria1) is composed of 430 spherical images with fast Y axis turns (up to 25 degrees between consecutive frames) and with translations of around 0.15 meters. Conversely, the second real sequence

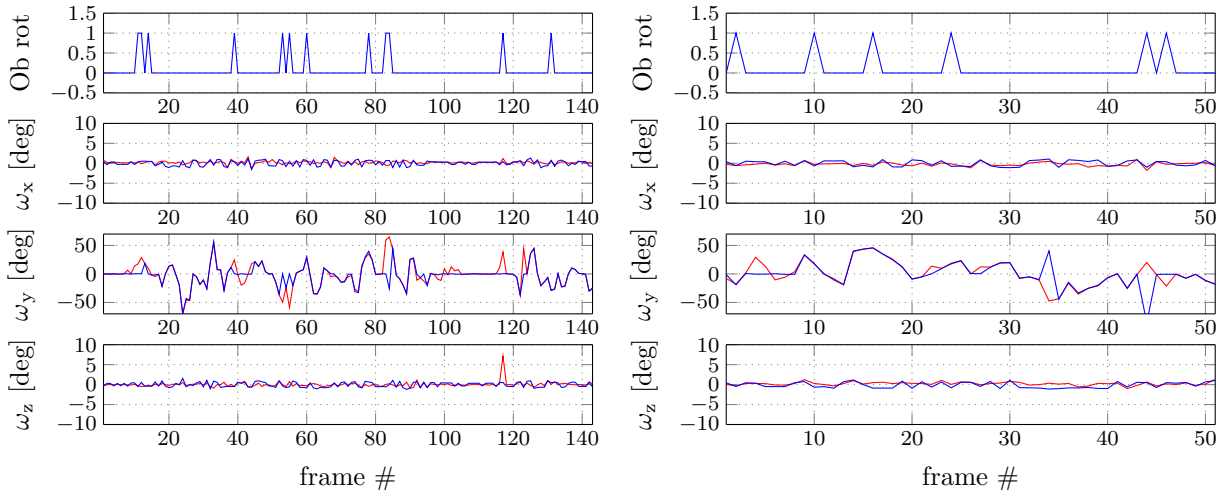

Figure 4.15 – Rotation estimation results for two different real sequences. The ground truth angles are the curves in red and the initialization curves are in blue. The graphics in the first column correspond to the sequence Inria1 and the second column to the Inria2. The first graphic at each column depicts the rotation observability index (which is one if the distribution is bi or multi-modal). See the text for details.

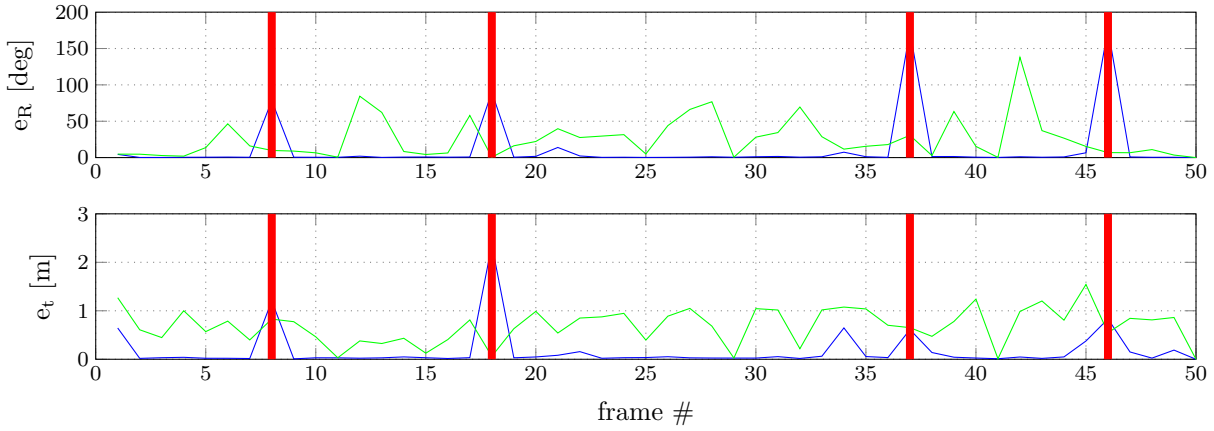

Figure 4.16 – Rotation and translation errors of the RGB-D registration without and with the initialization, for the trajectories shown in fig. 4.17. The registration is greatly improved by the initialization, as can be seen by the error curves with (in blue) and without the initialization. The red bars indicate the frames without co-visibility when the robot crossed the doors between the hall and offices.

(Inria2) is acquired with moderate rotations (up to 5 degrees) around the Y axis. To emulate large displacements, we selected a gap of 3 and 20 frames respectively. The rotation estimation was successful in 90% of cases having motions up to 70 degrees (see the left images of fig. 4.15 for Inria1) and up to 50 degrees in Inria2 (depicted in the right). These rotations can be seen in the angle ω_y in the left column of fig. 4.15 in the frame numbers 22 and 30. The translation estimate is however three times more sensitive to the noise than in the simulated sequences.

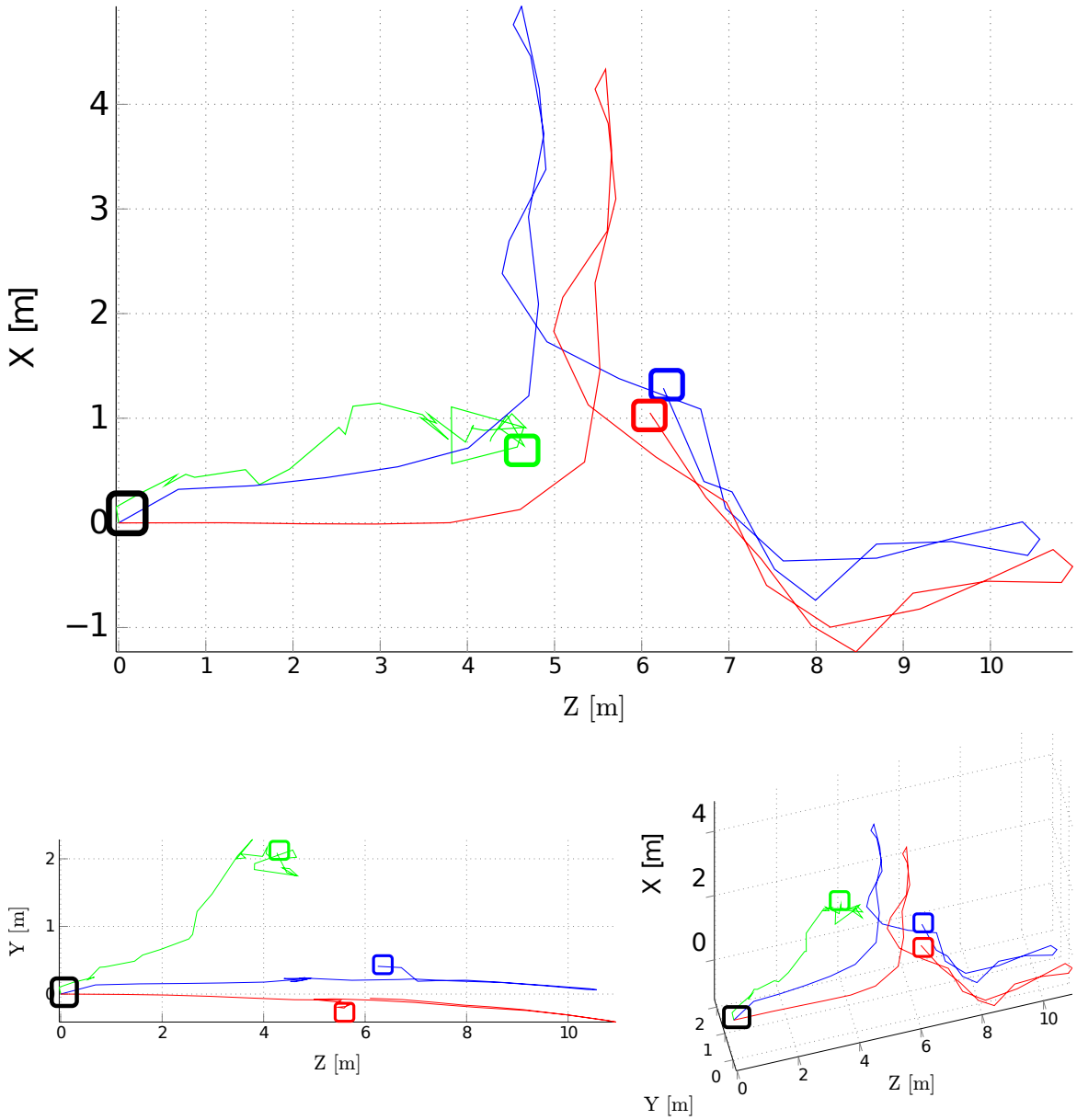

Figure 4.17 – Trajectories of direct RGB-D registration without (green) and with the initialization (blue) for the Inria1 sequence. The start point of the trajectories is indicated by the black box and the endpoint of each trajectory by the boxes with respective colors. The ground truth trajectory is depicted in red. For visibility, the plotted trajectories exclude the frames of door crossing. The pose estimation accuracy and the convergence were greatly improved when using the initialization.

4.4.3 Initialization of Direct RGB-D Registration

Finally, we use the pose estimation from normals in an initialization scheme to direct RGB-D registration, as described in section 4.3.1. The adaptive registration technique presented in chapter 5 is selected to assert the influence of the initialization in the registration. The maximum number of iterations is set to 20 per pyramid level. The initialization greatly improved the convergence of the registration, as it can be observed in the trajectories displayed

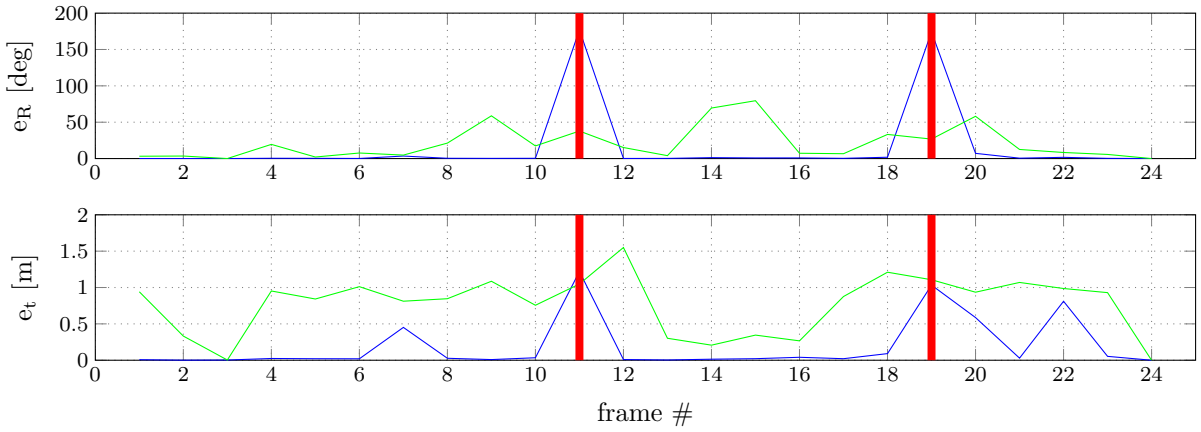

Figure 4.18 – Rotation and translation errors of the RGB-D registration without (in green) and with the initialization (in blue), for the trajectories shown in fig. 4.19. The registration is greatly improved by the initialization, as can be seen by comparing the errors and from the resulting trajectories. The red bars indicate the frames without co-visibility, when the robot crossed the doors between the hall and an office.

in figs. 4.16 and 4.17 from the real sequence Inria1. The registration without the initialization is depicted in green, with the initialization in blue and the approximative ground truth in red. The start point of the trajectories is indicated by the black box and the endpoint of each trajectory by the boxes with respective colors. The estimation failed only in frames without/minimal co-visibility, which is expected to happen because both initialization and registration are appearance-based techniques. These cases are indicated by the red bars in fig. 4.16 and happened while the robot was passing through doors, where the reference and current frames were located in opposite sides. Finally, the respective pose errors and trajectories using the frames from the Inria2 sequence are shown in figs. 4.18 and 4.19 without and with the initialization. As it can be noticed, the initialization greatly improved the convergence of the direct method in both sequences.

4.5 Conclusions

We presented in this chapter a decoupled rotation and translation estimation technique from large FOV depth images. First, a rotation rough estimation method is developed using the overlapping property of normal vectors between two views. This is performed thanks to a projector decomposition of the normal vectors in a general coordinate system. The technique does not assume any data pre-processing or "manual" segmentation (e.g., rejecting discontinuous depth regions). The translation is then directly derived from the rotation as a linear system of equations. We present some techniques to improve the conditioning of this system and a discussion about our assumptions and the limits of the method. This method does not assume any motion prediction or feature extraction/matching.

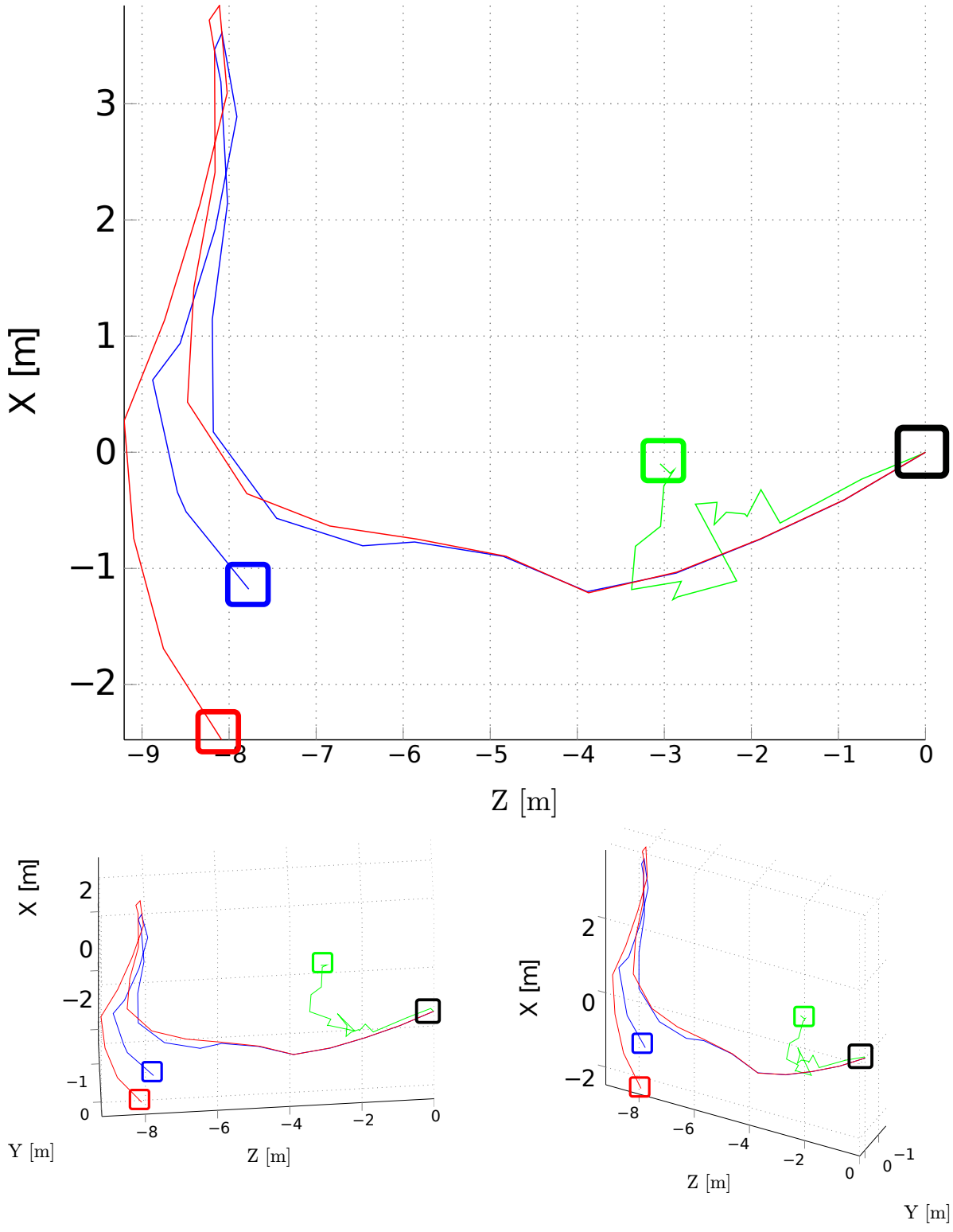

Figure 4.19 – Trajectories of direct RGB-D registration without (green) and with the initialization (blue) for the Inria2 sequence. The start point of the trajectories is indicated by the black box and the endpoint of each trajectory by the boxes with respective colors. The ground truth trajectory is depicted in red. For visibility, the plotted trajectories exclude the frames of door crossing. We shall remark the improvement in the accuracy of the pose computation when using the initialization.

Experiments are performed with simulated and real spherical sequences to the pose computation from normal vectors, showing an interesting compromise of accuracy and running time. Due to its efficiency and domain of convergence, the pose computation is used within an initialization scheme of a direct RGB-D registration method, where the accuracy and convergence of the registration was greatly improved. The initialization technique allowed the convergence with rotations of 170 degrees and translations of two meters in the indoor sequences. The pose estimation and registration only failed while registering frames without (or with minimal) co-visibility, which is expected to happen because both initialization and registration are appearance-based techniques. By last, we enforce the efficiency of the pose and initialization computation. The pose estimation algorithm runs, in a Matlab non-optimized code, with around 0.015 seconds for the rotation and 0.03 seconds for the translation (with Matlab 2012 in a laptop Intel Core i5-5300U CPU, 2.3 GHz and Ubuntu 14.04).

A natural extension of this technique is to explore other information sources to find the overlapped regions, as intensity/color. This could discriminate similar modes in the distributions. Moreover, the color information could also be used in a unidimensional search to improve the translation estimation in corridor-like environments. Another interesting research direction would be to combine this pose estimation in the Branch and Bound algorithm presented in [Yang et al., 2016]. From the presented results, we could expect to reduce the computational cost by replacing the ICP method in their formulation. We leave these considerations as future research.

Chapter 5

Adaptive Direct RGB-D Registration for Large Motions

Contents

5.1 Introduction

This chapter proposes a strategy to increase the basin of convergence of appearance-based (direct) RGB-D registration methods. The interest of direct methods is their accuracy. However, in general, direct image registration (e.g., [Comport et al., 2010, Tykkala et al., 2011, Kerl et al., 2013b, Silveira, 2014]) techniques assume high frame rate (small camera motions). The convergence of these methods depends on a number of parameters including: the noise in the photometric and geometric images, the scene configuration (photometric and geometric symmetries), the scene stationarity (i.e., without illumination changes or moving objects) and, of course, the camera motion.

In the past few years, the recent market of RGB-D commodity sensors opened new perspectives in terms of efficiency and robustness to perform tracking and mapping tasks [Newcombe et al., 2011a]. In this context, we are interested in exploring the complementary nature of the intensity and geometric images given by these sensors, in order to increase the basin of convergence and the convergence rate. In other words, to allow direct RGB-D approaches to consider moderate to large displacements, while ensuring nice convergence properties and accuracy. This is useful for a set of scenarios such as: high speed camera motions or low frame rate acquisition [1]. But also when performing model-based visual odometry in large-scale scenes, where the stored model is sparse due to storage limitations. A practical example is of performing real-time localization in a previous acquired sparse keyframe graph (e.g., as in [Meilland et al., 2015, Gokhool et al., 2015, Maier et al., 2015]). In these cases, the registration techniques have their performance challenged, being subjected to local minima or to small convergence rate (for instance, fig. 5.1 display two examples of frames with such conditions). This chapter addresses a contribution in this direction by leveraging information gathered from intensity images [Comport et al., 2010] and depth images (with ICP point-to-plane [Gelfand et al., 2003]), not only for improving ranking conditioning as in [Tykkala et al., 2011] but also accounting the properties of intensity and geometric cost terms.

The remainder is organized as follows: section 5.1.1 reviews recent related works. In sections 5.2 and 5.3, we introduce the basic classical method of RGB-D registration and our adaptive approach. We present experimental results in section 5.4 for indoor (simulated and real) and outdoor sequences from the KITTI VO/SLAM dataset [Geiger et al., 2012] considering both perspective and spherical images. Finally, in section 5.5, we draw conclusions and highlight possible future improvements.

1. The conditions of high speed and low frame rate are not technically equivalent because high speed creates blurring and distortions in the images. With global shutter cameras and "moderate" motions, these effects can be neglected and the image formation follows the central projection assumption. Otherwise adapted formulations using rolling shutter camera models should be considered (e.g., [Meilland et al., 2013, Kerl et al., 2015, Saurer et al., 2015]).

5.1.1 Main Related Works

Large pixel displacement estimation is an active research area in the optical flow community [Brox and Malik, 2011, Timofte and Gool, 2015, Muller et al., 2011, Braux-Zin et al., 2013]. These works compute the pixel displacements (a dense flow field) considering different constraints/regularizations inside a variational optimization framework [Chambolle, 2004]. [Brox and Malik, 2011] proposed a modified energy cost for dense optical flow estimation combining feature and appearance-based approaches for a trade-off between accuracy and convergence domain. Their energy formulation combines appearance energy terms (color, gradient) with feature matching terms (e.g., using SIFT, SURF, ORB descriptors). The method is embedded in a multi-resolution scheme, where the weight of the feature's terms are progressively reduced while increasing the resolution. [Braux-Zin et al., 2013] extended this work, enabling a wider class of features to be integrated in the cost, such as line segments. In the same context, the work of [Muller et al., 2011] described a scene flow estimation method more robust to weakly textured and fast moving image regions. The core of their algorithm is to perform an initialization of the flow computed from co-visible and static pixels. Instead of using the disparity values directly, the authors proposed to use the flow computed from static objects – using an independent estimate of the depth from stereo SGBM [Hirschmuller, 2008] and from the motion. The authors called this approach of modified total variation (MTV) and combined this strategy with the formulation of [Brox and Malik, 2011] for handling image regions with large motions. In short, their formulation included stereo, feature matching constraints, spatial and temporal predictions in the variational scheme. In order to increase the basin of convergence of direct methods, the authors of [Hadj-Abdelkader et al., 2008] proposed to compute a pose initialization with a feature-based registration technique. The core of their method is an adapted feature extraction that is less sensitive to image distortions induced by large FOV sensors (such as catadioptric, spherical and fisheye sensors). In summary, the previous approaches are not suitable in our context, because we aim to maintain the direct estimation concept, i.e., without

Figure 5.1 – Typical frames with large displacement motions and challenging conditions: occlusions and dynamic objects in the indoor case (left and center-left figures) and varying lighting conditions and poor geometric stereo estimation for an outdoor frame (center-right and right figures).

feature extraction and matching.

Naturally, this work is also closely related to direct RGB-D motion estimation techniques (e.g., [Kerl et al., 2013b, Korn et al., 2014, Kerl et al., 2015, Munoz and Comport, 2016a]), in particular to [Morency and Darrell, 2002, Tykkala et al., 2011, Munoz and Comport, 2016b]. An important issue raised in these works is the scaling of the geometric and photometric cost terms for ensuring nice convergence properties, i.e., to weight the influence of the intensity and geometric appearance-based errors. In [Morency and Darrell, 2002], the scaling factor follows a sigmoid function considering the reprojection error of the depth images. Although sharing a similar framework and initial conclusions, we propose two different scaling functions. The first is assuming that the increments of pose are smaller when nearer the solution. The second is based on the conditioning of the error terms, which is of easier tuning compared to the first one. The second scaling is also capable of dealing with cross-peak optimization instabilities that can appears while shaping the cost function. In [Tykkala et al., 2011], the scale factor transforms the geometric error (in meters) to pixels using the ratio of the median values of $\boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{D}}$. This metric ensures better ranking conditions (e.g., in cases of non-textured regions) with similar convergence rate, but fails to handle basic cases of bimodal pixel/depth distributions and does not consider the complementary properties of both intensity and depth images. Furthermore, the intensity cost "dominates" the convergence properties using this scheme, as in our approach. Similarly, [Korn et al., 2014] combines intensity and geometric terms using an heuristic constant scaling factor. Recently, [Munoz and Comport, 2016a, Munoz and Comport, 2016b] proposed an RGB-D registration using a point-to-plane error in higher dimensions that is invariant to the scaling factor. Their formulation uses the notion of hyperplanes (planes in $\mathbb{R}^n, n > 3$). However, the computation of the normals in higher dimensions is computationally expensive, in the order of seconds, even for low resolution images.

5.1.2 Contributions

The contribution of this chapter is an adaptive scaling of the intensity and depth costs that improves the convergence of direct RGB-D registration. We show that the intensity and depth cost terms display different convergence properties for small and large motions (for instance, a typical example is given in figs. 5.2 and 5.3). Two possible scaling functions (activations) are proposed to update the scaling: the pose evolution and the relative condition number. The former is related to the scaling presented in [Morency and Darrell, 2002]. Both scalings have faster registration convergence using simulated and real RGB-D sequences than [Tykkala et al., 2011]. Extensive RGB-D registration experiments for different scenarios and sequences shows a significant improvement of the basin of convergence and the convergence speed.

It is worth noting that this approach is valid for moderate/large motions, but still with a basin of convergence smaller than feature-based methods (see section 3.2.1 of chapter 3 for a short survey). In extreme cases, such as while performing loop closure, we are likely to

converge to local minima and therefore other techniques might be used instead, e.g., feature-based methods or global registration techniques.

5.2 Classic RGB-D Registration

In this chapter, a frame $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}\}$ is composed of a grayscale image $\boldsymbol{\mathcal{I}} \in [0,1]^{m \times n}$ of the RGB and of the depth $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{m \times n}$ image. The sensor projection models of interest are the perspective and the spherical, as described in chapter 2. In the later, the images are projected in the unit sphere using the equirectangular projection as detailed in section 2.2. Notice that the spherical representation can generalize other wide FOV sensors, obeying the central projection assumption, from a calibration procedure [Puig et al., 2012].

The direct image registration consists on finding iteratively the pose $\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})$ between a reference $\boldsymbol{\mathcal{S}}^*$ and a target frame $\boldsymbol{\mathcal{S}}$ from the images appearance, i.e., from the photometric and geometric errors:

$$e_I(\mathbf{p}, \mathbf{x}) = \boldsymbol{\mathcal{I}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}) \tag{5.1}$$

$$e_D(\mathbf{p}, \mathbf{x}) = \lambda_D(\hat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{n}^*(\mathbf{p}))^T(\mathbf{P}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) - \hat{\mathbf{T}}\mathbf{T}(\mathbf{x})\mathbf{P}^*(\mathbf{p})). \tag{5.2}$$

Where: $\hat{\mathbf{T}}$ is an initial pose guess; $\mathbf{n}^*$ the normal surface vector calculated at the reference frame; $\mathbf{P}$ is the 3D point using the depth image and the inverse camera projection model; and λ_D is a tuning parameter for scaling the error terms. The eq. (5.1) can be seen as a classical optical flow constraint equation (OFCE), within the hypothesis of Lambertian surfaces, and (5.2) is equivalent to a flow point-to-plane ICP, both assuming predominant static scenes. To ensure these assumptions, robust M-estimators (denoted as ρ) are applied for mitigating the influence of outliers [Zhang, 1995], therefore, reducing the effects of occlusions, moving objects, changes of illumination and interpolation errors during the estimation.

The classic RGB-D registration consists of minimizing jointly (5.1) and (5.2) in a convex cost as:

$$C(\mathbf{x}) = \min_{\mathbf{x}} \left(\sum_{\mathbf{p}} \rho_I(e_I(\mathbf{p}, \mathbf{x})) + \sum_{\mathbf{p}} \rho_D(e_D(\mathbf{p}, \mathbf{x})) \right). \tag{5.3}$$

The image warping and the pose follows the same parametrization exposed in sections 3.3.1

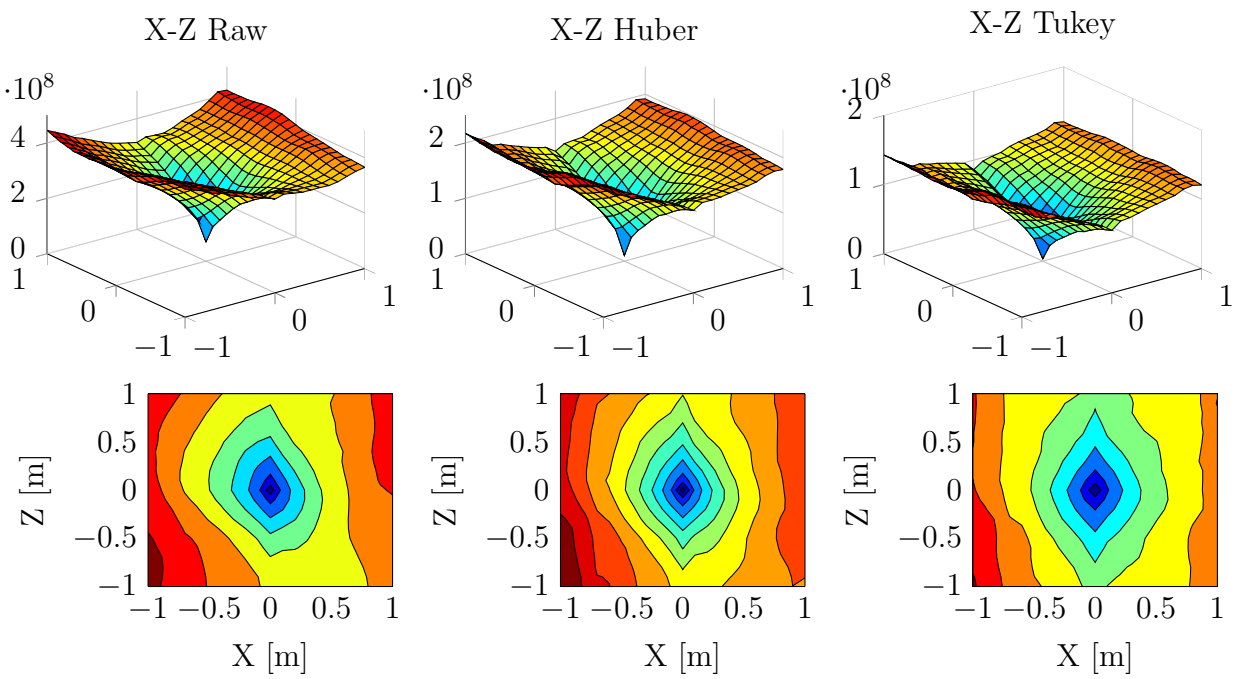

Figure 5.2 – Intensity cost function for the X-Z DOFs (top row) for different robust functions: Euclidean norm (Raw), Huber M-estimator (Huber) and Tukey's bisquare (Tukey). The bottom row depicts the level curves of the respective costs of the top row. The attraction domain of the costs are similar for both robust estimators. The intensity cost is more discriminant near the solution than the ICP cost depicted in fig. 5.3.

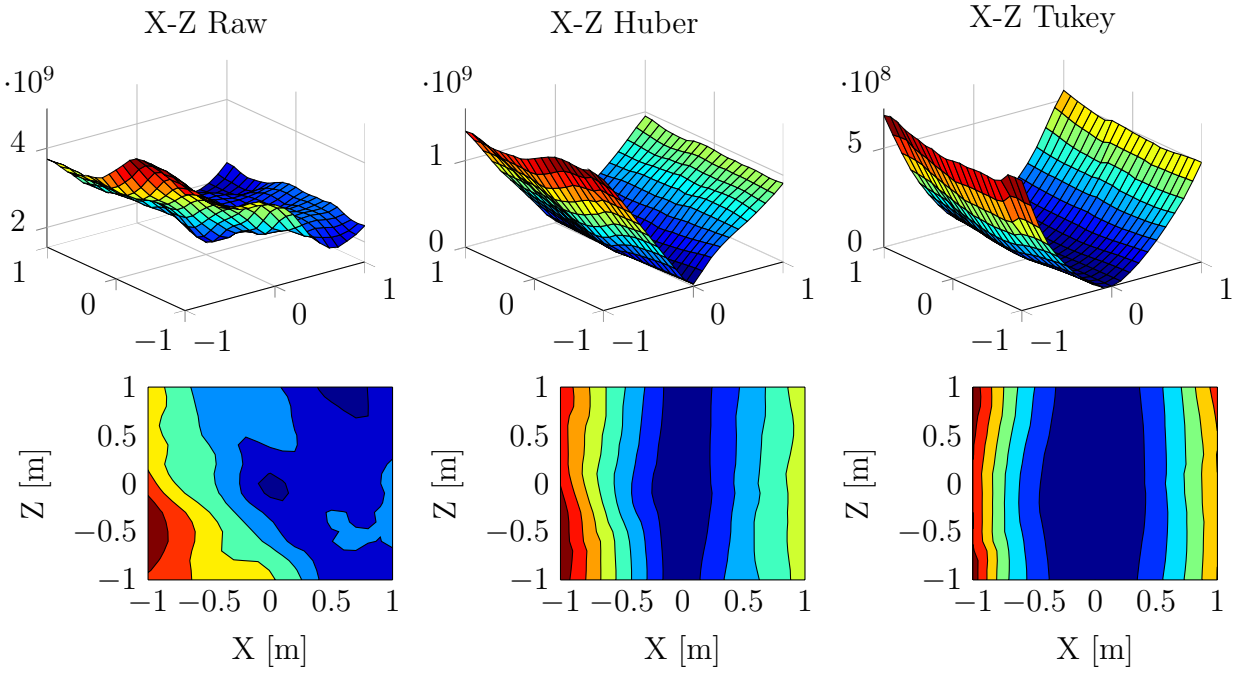

Figure 5.3 – ICP cost function for the X-Z DOFs (top row) for different robust functions: Euclidean norm (Raw), Huber M-estimator (Huber) and Tukey's bisquare (Tukey). The bottom row depicts the level curves of the respective costs of the top row. The attraction domain of the costs are similar for both robust estimators. The ICP point-to-plane cost is flatter near the solution than the intensity error depicted in fig. 5.2.

and 3.3.3.1:

$$\begin{aligned} &\bullet \text{ Perspective: } w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})) = w(w(\mathbf{p},\mathbf{T}(\mathbf{x})),\hat{\mathbf{T}}) = \|\boldsymbol{\mathcal{D}}(\mathbf{p})\mathbf{K}\hat{\mathbf{R}}\mathbf{R}\mathbf{K}^{-1}\mathbf{p} + \mathbf{K}(\hat{\mathbf{R}}\mathbf{t} + \hat{\mathbf{t}})\|_P \\ &\bullet \text{ Spherical: } w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})) = w(w(\mathbf{p},\mathbf{T}(\mathbf{x})),\hat{\mathbf{T}}) = \Pi_S\left(\|\hat{\mathbf{R}}\mathbf{R}\boldsymbol{\mathcal{D}}(\mathbf{p})\Pi_S^{-1}(\mathbf{p}) + \hat{\mathbf{R}}\mathbf{t} + \hat{\mathbf{t}}\|_S\right) \end{aligned} \tag{5.4}$$

Denoting: Π_S is the spherical equirectangular projection; $\mathbf{K}$ is the intrinsic pinhole camera matrix; $\|\ \|_P$ and $\|\ \|_S$ are the perspective and spherical normalizations. It is worth noting that selecting a large λ_D ($\lambda_D >> 1$) in (5.3) is, in the limit, equivalent to a direct point-to-plane ICP method, while small λ_D ($\lambda_D \approx 0$) corresponds, in the limit, to a classical direct intensity based registration. To increase the basin of convergence, the minimization of the cost 5.3 is often performed within a multi-resolution framework, e.g., performing smoothing with a Gaussian kernel and sub-sampling operations such as in [Burt and Adelson, 1987] to build a Pyramid of images with different resolutions. The optimization starts in the smallest resolution (pyramid at level n) to the higher image resolution (level 1).

5.2.1 Convergence of Intensity and Geometric Registration

We observed in both simulated and real sequences that the intensity and geometric terms have distinct convergence properties. Although the convexity analysis of the cost terms in (5.1) and (5.2) cannot be established in general, the intensity RGB term has often slower convergence than the ICP point-to-plane cost, but its locally more accurate (for instance, figs. 5.2 and 5.3 depict two typical examples). This agrees with the findings of [Morency and Darrell, 2002] in face tracking tasks and with the convergence differences of position-based visual servoing and image-based visual servoing [Chaumette and Hutchinson, 2006]. Of course, other factors such as for instance, the choice of the robust estimator ρ and the approximation of the Hessian in eq. (5.3) can model the shape of the costs and the trajectory undertaken by the optimized pose parameter $\mathbf{x}$ during the minimization of $C(\mathbf{x})$.

For illustration, we present typical shapes of the RGB and ICP cost terms for two DOF in figs. 5.2 and 5.3 and for three DOF (two translations and one rotation) in fig. 5.4 using two frames in the Sponza Atrium model dataset. As can be noticed in figs. 5.3 and 5.4, the ICP point-to-plane is flatter than the RGB term for small interframe displacements, meaning that ICP is less discriminant in the vicinity of the solution. Furthermore, due to the scene symmetry along the Z axis (corridor-like environment), the convergence rate is likely to be slow following this DOF with symmetry (see fig. 5.4 bottom right level plot and fig. 5.3). Conversely, the geometric error component (second row in fig. 5.4) is more discriminant than the intensity cost (first row) when farther from the solution, as shown by the slope of the geometric cost in fig. 5.3.

This effect is not depending to the choice of the robust M-Estimator, as described by the experiment depicted in figs. 5.2 and 5.3 for the Euclidean, Huber or Tukey functions and of the

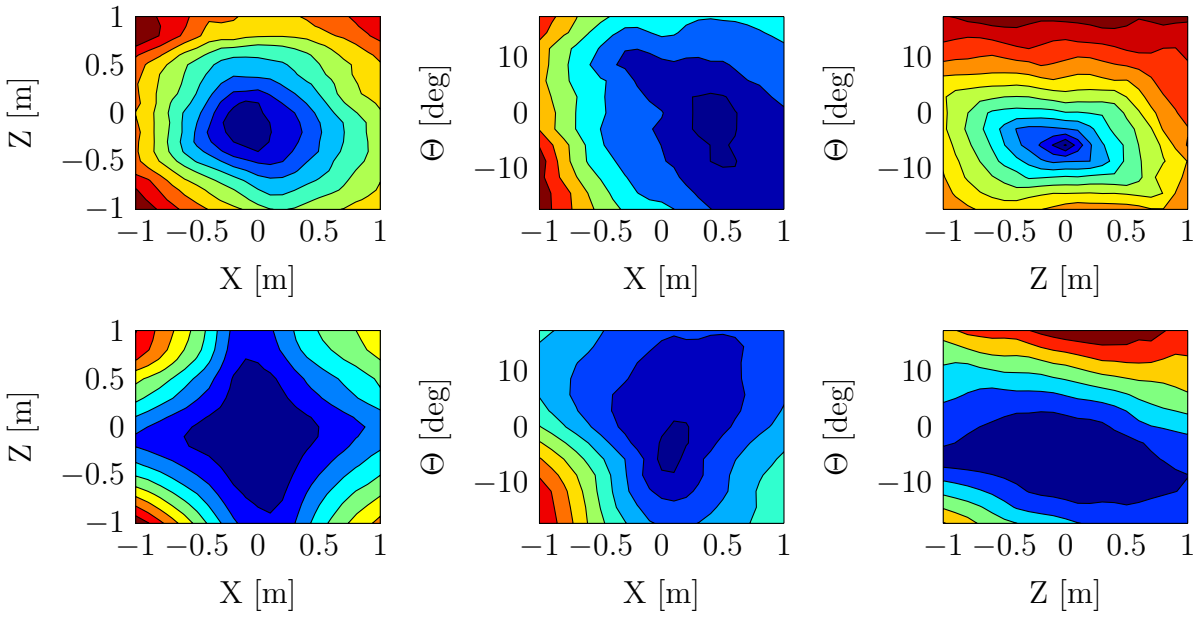

Figure 5.4 – Intensity RGB level curves (first row) and ICP ploint-to-plane (second row) for a typical corridor frame at the Sponza Atrium model. The costs are evaluated in the simplified case of three DOF (one rotation and two translations) and the corresponding level curves are from the surface intersection of $C(\mathbf{x})$ with the secant planes: $\mathbf{x} = [x\ 0\ z\ \mathbf{0}_{(1\times3)}]^T$ (left column), $\mathbf{x} = [x\ 0\ 0\ \theta\ 0\ 0]^T$ (middle) and $\mathbf{x} = [0\ 0\ z\ \theta\ 0\ 0]^T$ (right column). The ICP point-to-plane cost is flatter near the solution. Please see the text for details.

Hessian approximation (e.g., the gradient, Gauss-Newton or ESM). Interestingly, the intensity cost "dominates" the convergence rate using the combination of the error terms such as in [Tykkala et al., 2011]. An example of the registration improvement by taking into account the costs properties is shown in fig. 5.5 using two frames of the KITTI VO/SLAM dataset.

5.3 Adaptive Formulation

As stated previously in section 5.2, a main concern with direct methods is about their convergence, since only local properties are settled from eqs. (5.1), (5.2) and (5.3). We aim to explore the complementary aspects described in section 5.2.1, in terms of convergence, by using a modified cost function, where the geometric term prevails in the first iterations, while the intensity data term dominates in the finer increments. Instead of setting a constant scaling, our adaptive RGB-D registration approach is based on the classic RGB-D strategy combined with an activation scaling $\mu(\mathbf{x})$:

$$\tilde{C}(\mathbf{x}) = (1 - \mu(\mathbf{x})) \sum_{\mathbf{p}} \rho_I(e_I(\mathbf{p}, \mathbf{x})) + \mu(\mathbf{x}) \sum_{\mathbf{p}} \rho_D(e_D(\mathbf{p}, \mathbf{x})) \tag{5.5}$$

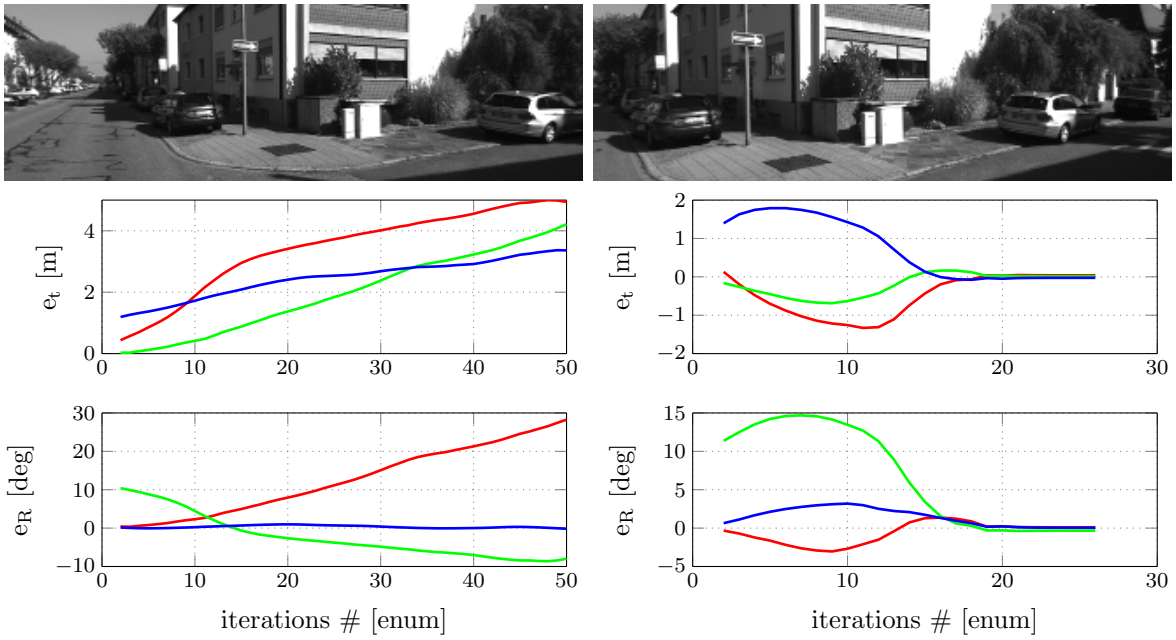

Figure 5.5 – Pose error evolution while registering frames 000105 and 000108 of the sequence 00 from the KITTI dataset. The frames are shown in the top row. The pose error evolution for the classic RGB-D with [Tykkala et al., 2011] is shown in the left column and the adaptive RGB-D in the right column. The translation error for each DOF are depicted in second row (errors in meters) and the rotation DOF errors are depicted in the third row figures (in degrees), where the convergence in this case was successful using the adaptive formulation.

where the respective Jacobians are scaled versions of the Jacobians from the original formulation in (5.3):

$$\tilde{\mathbf{J}} = [\sqrt{(1-\mu(\mathbf{x}))}\mathbf{J}^I \quad \sqrt{\mu(\mathbf{x})}\mathbf{J}^D]^T. \tag{5.6}$$

Based on the framework in (5.5) and on the convergence properties described in section 5.2.1, we are able to design the scaling parameter $\mu(\mathbf{x})$ as an activation function [2] for different camera motions. In the following two subsections, we describe the design of two plausible activation functions according to the expected photometric and geometric information gain.

5.3.1 Activation with Pose Evolution

Supposing regular cost functions with a quadratic shape, the derivatives (consequently the pose increments) are bigger when farther from the optimal pose minimizing (5.3) and smaller when near the optimal pose. A natural candidate activation function $\mu(\mathbf{x})$ in this context is the smoothed step depending on the size of the pose increments $\mathbf{x}$ along the minimization of

2. We denote the scaling factor as activation function due to its algebraic properties in the cost (5.5), similarly to the activation function connotation used in perceptrons of neural networks.

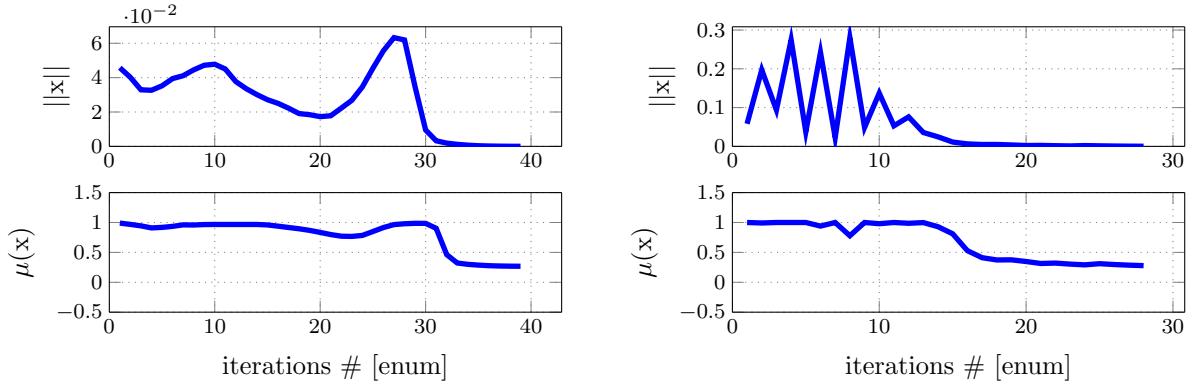

Figure 5.6 – Activation adaptive function $\mu(\mathbf{x})$ (5.7) while performing registrations in the KITTI outdoor dataset sequence 00. The tuning parameters are given in Table 5.1. Notice that the norm of the pose increments are not monotonically decreasing (top graphs).

(5.5):

$$\mu(\mathbf{x}) = k_1 / \left(1 + \exp(-k_2(\|\mathbf{x}\|_2 - k_3))\right) \tag{5.7}$$

with $0 < k_1 < 1$ and $(k_2, k_3) > (0, 0)$. Please remark that selecting a high value to k_2 and small k_3 is, in the limit, equivalent to perform a sequential independent ICP and intensity RGB registration, i.e., in cascade but loosing the complimentary properties of both terms. Therefore, this activation is particularly sensitive to the tuning parameters k_2 and k_3, which can induce oscillations (as known as cross-peak instabilities) by transforming the original cost (5.3) in a non-convex function. Two typical examples of this activation function along a registration in the KITTI dataset are given in fig. 5.6.

5.3.2 Activation with Relative Conditioning

A second strategy is to design $\mu(\mathbf{x})$ from the costs relative behavior along the minimization. It is worth noting the well defined minimum displayed by the intensity cost, as show in the example described in fig. 5.3. The idea is that the relative variation of the costs (which is encoded as the relative conditioning number) could "detect" when the optimization is in the vicinity of the solution, i.e., when the ICP cost is less discriminant than intensity. Hence, a plausible adaptive function candidate is:

$$\mu(\mathbf{x}) = \begin{cases} k_1 + k_2, & \text{if } \ \text{cond}_{\mathbf{x}}(C_I(\mathbf{x}))/\text{cond}_{\mathbf{x}}(C_D(\mathbf{x})) < k_3 \\ k_1, & \text{otherwise.} \end{cases} \tag{5.8}$$

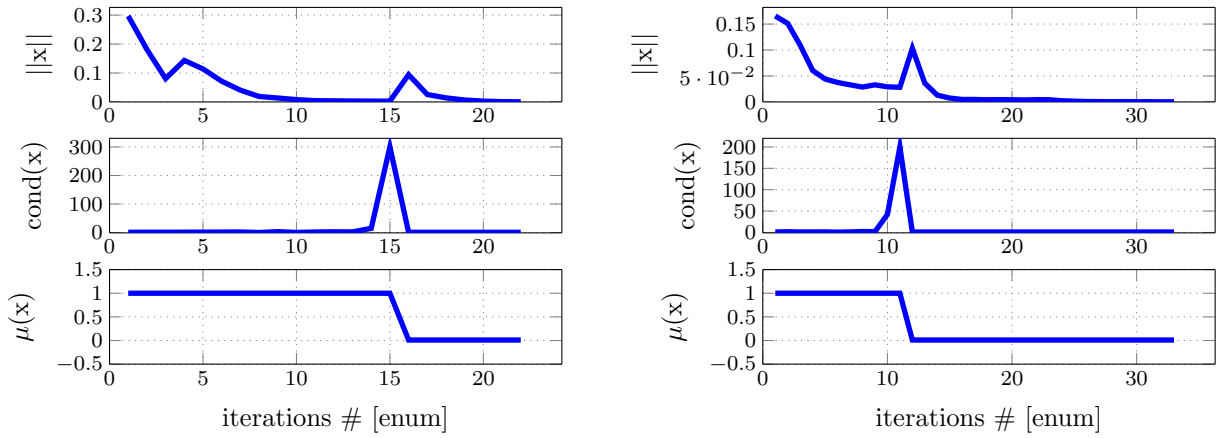

Figure 5.7 – Activation adaptive function $\mu(\mathbf{x})$ while performing registration in the KITTI outdoor dataset in two different areas (frames' numbers 5 and 100) of sequence 00. The left column corresponds to a scene where the convergence is slower (corridor/canyon-like environment) and the right column is of frames affected predominantly by a rotation. The conditioning criteria (5.8) is easily detectable for both cases.

where $0 < k_1 \leq k_1 + k_2 < 1$, k_3 is large ($k_3 >> 1$) and cond is an approximation of the relative condition number of the RGB (C_I) and ICP (C_D) cost functions such as:

$$\text{cond}_{\mathbf{x}}(C(\mathbf{x})) = \left| \frac{C(\mathbf{x}_0 \oplus \mathbf{x}) - C(\mathbf{x}_0)}{C(\mathbf{x}_0)} \right|_1 / \frac{\|\mathbf{x}\|_2}{\|\mathbf{x}_0\|_2}. \tag{5.9}$$

The minimal pose parametrization in (5.9) is given by $\mathbf{x}_0 = \text{vex}(\log(\hat{\mathbf{T}}))$ and $\oplus$ is a composition operator such as $\mathbf{x}_0 \oplus \mathbf{x} = \text{vex}(\log(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})))$ [3]. We show in fig. 5.7 typical activation functions using the KITTI VO/SLAM dataset [Geiger et al., 2012]. The parameters of each activation are detailed in Table 5.1. Two activation examples are displayed in fig. 5.7, where this activation criteria detects correctly the sensitivity of the costs, whilst being of simple tuning.

5.3.2.1 Robust Estimators

Finally, we adopted the Huber robust function for ρ_I, ρ_D for ensuring convexity properties when farther from the solution [Zhang, 1995]. To avoid outliers influence, the robust function is switched to Tukey's bisquare when in the vicinity of the optimal motion (i.e., when the conditioning in (5.8) is large). The respective Jacobians and details about the optimization are given in the Appendix A.

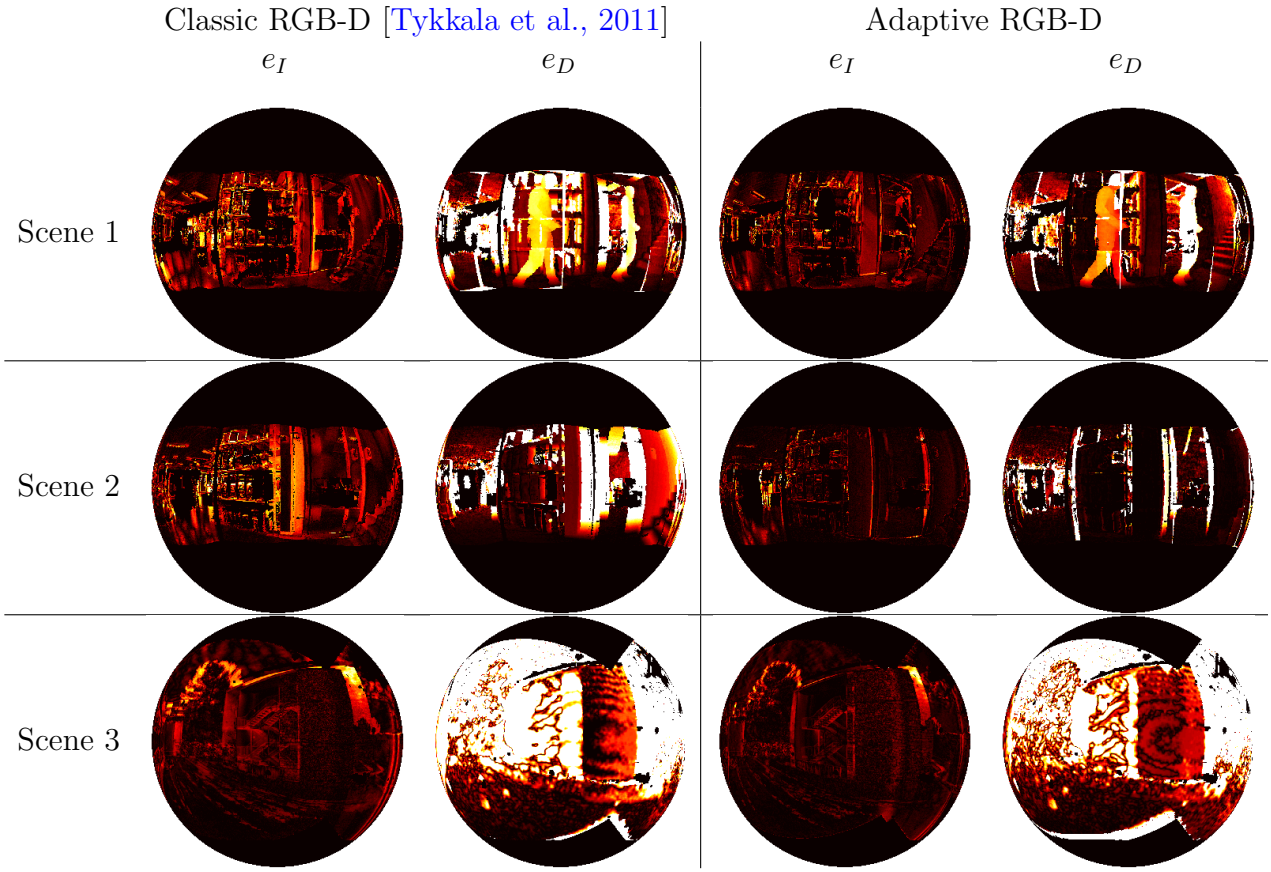

Figure 5.8 – Intensity (e_I) and geometric (e_D) errors between spherical RGB-D frames with large displacements for three distinct indoor and outdoor sequences. The colors encode the error absolute value (hot colors indicate bigger errors than cold colors). The left column is the classic RGB-D cost and the last column the adaptive one. Scene 1 corresponds to the registration between the two left frames shown in fig. 5.1 from the sequence whose trajectory is displayed in fig. 5.10. Scene 2 corresponds to a registration from the trajectory depicted in fig. 5.11. Finally, Scene 3 is an outdoor sequence which frames are depicted at right of fig. 5.1.

5.4 Experiments and Results

In this section, we evaluate the adaptive registration in a set of sequences acquired in indoor and outdoor scenes using perspective and spherical RGB-D sequences. We consider the average of the rotation relative error (RRE), translation relative error (TRE) and number of iterations for convergence as quantitative metrics. A qualitative analysis is also done using the intensity and depth errors from the view related to the estimated pose. Unless specified, the term "adaptive RGB-D" corresponds to the cost (5.5) using (5.8). Among the possible direct methods for estimating the pose from RGB-D images (e.g., [Tykkala et al., 2011, Newcombe et al., 2011a, Kerl et al., 2015]), we select the method of [Tykkala et al., 2011] for comparison. The reason is twofold. First, it has similar structure and same computational complexity.

3. The expression $\mathbf{x}_0 \oplus \mathbf{x}$ can be computed explicitly, without passing by $\mathbb{SE}(3)$ group, via the Baker-Campbell-Hausdorff (BCH) formula [Li and Hartley, 2007, Quiroga et al., 2014].

Table 5.1 – Parameters in the activation functions.

	Parameters	Typical Range
Meth. 1 [Tykkala et al., 2011]	$\lambda_D = med(\boldsymbol{\mathcal{I}})/med(\boldsymbol{\mathcal{D}})$	$\lambda_D \in [5, 50]$
Adapt. 1 (5.7)	$\lambda_D = 1$, $k_1 = 1 - 10^{-5}$, $k_2 = 100$, $k_3 = 0.001$	$\mu \in [0, k_1]$
Adapt. 2 (5.8)	$\lambda_D = 1$, $k_1 = 10^{-5}$, $k_2 = 1 - 10^{-2}$, $k_3 = 30$	$\mu \in \{k_1, k_1 + k_2\}$

Second, it does not perform bundle adjustment or loop closure as in other state of art methods such as [Newcombe et al., 2011a, Kerl et al., 2015].

5.4.1 Implementation Aspects

The registration is assumed to have converged, either to a global or to a local minimum, when the norm of pose increments $\mathbf{x}$ are bellow a fixed threshold in successive iterations (10^{-5} for the rotation and 10^{-3} for translation). The selected values for the parameters λ_D and in the activation functions (5.7) (5.8) are described in Table 5.1. These parameters are kept constant during all the experiments.

We start in a fixed resolution for evidencing the differences between the classic RGB-D and the adaptive formulation. Unless specified, the maximum number of iterations is 50. To emulate different motion speeds, only a sub-set of the frames is picked up using a sub-sampling of the sequences (gaps), e.g., a gap of 10 frames corresponds to compute the relative pose between the frame pairs (1,11), (11,21), ..., $(i, i + \text{gap})$.

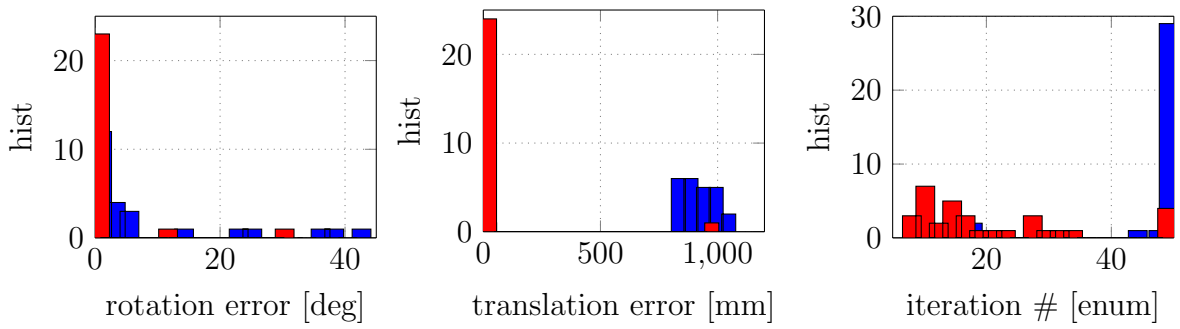

Figure 5.9 – Rotation error (degrees), translation error (millimetres) and number of iterations using a fixed image resolution for the simulated testbed dataset with gap of 10 frames. The results considering the classical RGB-D registration model is presented in blue, while the adaptive formulation using the conditioning is depicted in red. As it can be noticed, the accuracy and convergence rate are substantially improved when exploiting the activation factor in the cost.

5.4.2 Spherical Simulated Sequence

At first, we evaluate the approach in controlled conditions using 500 RGB-D spherical synthesized images from the Sponza Atrium model. The inter-frame distances are in average of 0.15 meters and of 4 degrees in rotation. The registration was performed for sub-sampling of 5, 10 and 15 frames and the results are synthesized in Table 5.2 and fig. 5.9 shows the pose errors and the number of iterations for a gap of 10 spheres. The convergence was achieved even in cases considering translations and rotations of around 2.5 meters and 60 degrees (a result that was also observed and discussed in the initialization described in chapter 4). The adaptive approach has a better performance in the accuracy and in the computation effort for all the considered gaps, and failed to converge in less than 10% of trials considering a gap of 15 frames. The convergence failed mostly when the reference scene was almost completely occluded in the target frame (e.g., in corridor 90 degrees turns) and they were expected because the frames have no co-visible information.

5.4.3 Spherical Indoor and Outdoor Real Sequences

The spherical real sequences were acquired using the two spherical RGB-D rigs [Fernandez-Moral et al., 2014, Meilland et al., 2015] described in the introductory chapters. The indoor images are from the hall and offices of the Inria building using the eight Asus sensor rig, while the outdoor sequence was acquired in the Inria campus using the stereo spherical rig. Unfortunately, we do not have ground truth of the poses in these sequences. Hence, a more qualitative analysis is done to evaluate the performance of the methods. We depict three registration experiments in fig. 5.8 for the indoor and outdoor sequences. The selected frames contain large motions, occlusions and moving objects. The respective appearance error in the intensity and depth images are shown after 20 iterations. These errors are considerably smaller for both indoor and outdoor examples using the adaptive scaling.

Subsequently, we show some trajectories in figs. 5.10 and 5.11 after applying the registration in the indoor sequences. The maximum number of iterations was set to 150 iterations for the registration with fixed resolution, and of 20 iterations per pyramid level in the multi-resolution. Using a sub-sampling of five frames, the method did not converge in only 9% of the trials as shown in fig. 5.10 for a fixed resolution. We can observe that the adaptive formulation has a

Table 5.2 – Quantitative results using the simulated spherical indoor sequence in a **fixed resolution**: average RRE[deg]/RTE[mm]/iterations.

	Gap = 5	*Gap = 10*	*Gap = 15*
Meth. 1 [Tykkala et al., 2011]	3.67/423/47.3	7.80/1104/48.4	11.7/1520/48
Adapt. 1 (5.7)	0.68/96.4/31.2	1.11/466/32.9	2.17/833/34.8
Adapt. 2 (5.8)	**0.03/88.6/31**	**0.04/182/26.5**	**0.05/523/20.7**

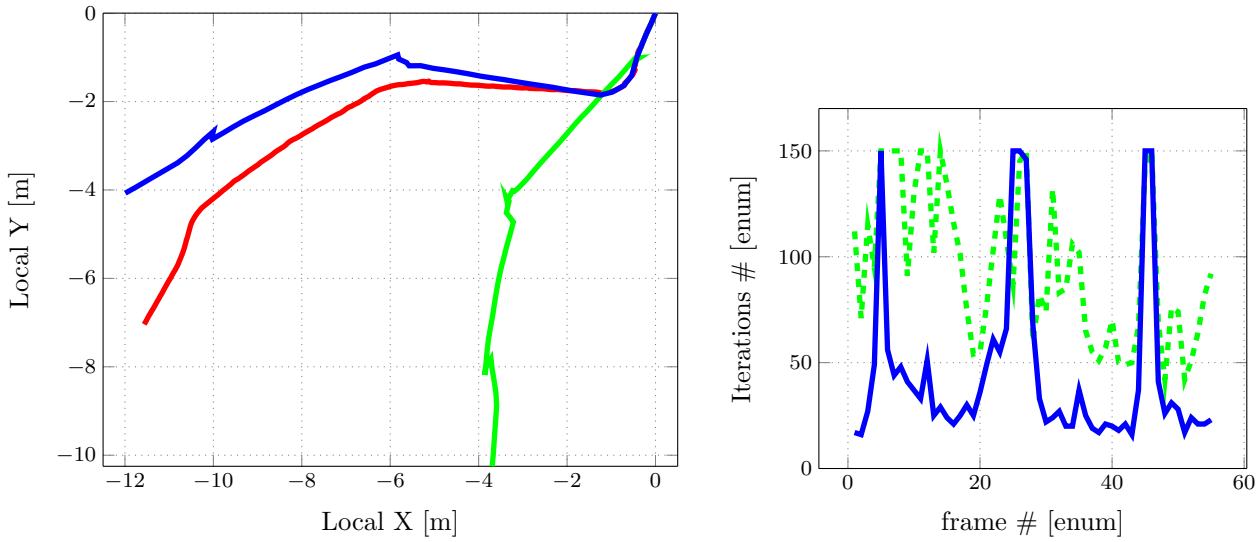

Figure 5.10 – Trajectory comparison for classic RGB-D [Tykkala et al., 2011] and adaptive formulations using the indoor spherical real sequence (left) and number of iterations (right). The registration considering the classical RGB-D is presented in green, while the adaptive formulation using the conditioning is in blue and the approximative ground truth trajectory in red. The accuracy and convergence rate are substantially improved when exploiting the activation scaling (in blue) for 87% of the frames.

better performance with a reduced number of iterations. Similar observations were obtained from the trajectories produced with multi-resolution, as shown in fig. 5.11. Some drift is present in both trajectories since we do not perform loop closure. However, by observing some corresponding areas of the environment, the adaptive method produced a more consistent trajectory.

5.4.4 KITTI Outdoor Perspective Sequence

We also provide results using a perspective stereo sequence from the KITTI Visual Odometry/SLAM benchmark. The depth information was pre-computed by rectifying the stereo perspective images and using ELAS [Geiger et al., 2010] for the disparity computation. It is a challenging dataset because the scene is mainly semi-structured (roads in an urban area) and with a travel speed of up to 60 km/h. Hence we selected smaller gaps of one, two and of three frames. Observe that in the outdoor scenario the overlapping regions are much sparser because only the road plane is the persistent overlapping surface. Furthermore, the perspective camera model restricts co-visibility. The respective errors are displayed in Table 5.3 for a fixed resolution, showing that the adaptive formulation surpassed again the other registration techniques. The first 200 meters of the KITTI sequence 00 with the higher resolution images are portrayed in the left graphic of fig. 5.12.

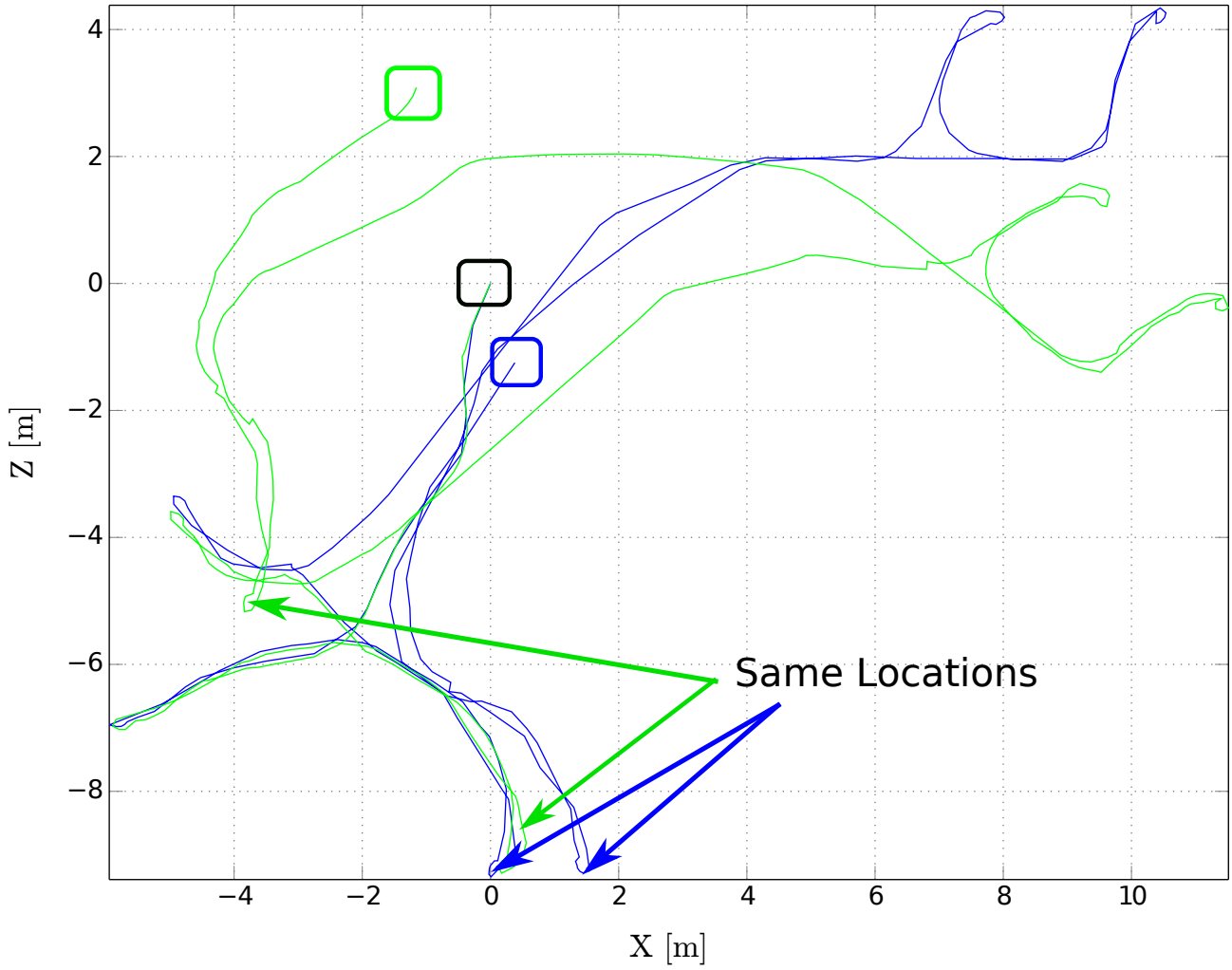

Figure 5.11 – Inria sequence mapping using the classic RGB-D (in green) and with the adaptive registration technique (in blue), both using multi-resolution. The trajectories started in the black box marker and ended in the respective colored boxes. A revisited place in the scene is indicated by the blue and green arrows. Observe that the drift using our method is much smaller.

5.4.4.1 Multi-resolution

Lastly, we combine the adaptive formulation with a multi-resolution Gaussian pyramid of four levels (the higher the level, the smaller the image resolution is) to assets the efficiency

Table 5.3 – Quantitative metrics using the KITTI outdoor sequence in a **fixed resolution**: average RRE[deg]/RTE[mm]/iterations.

	Gap = 1	*Gap = 2*	*Gap = 3*
Meth. 1 [Tykkala et al., 2011]	0.51/219/45.6	1.83/1071/49	2.75/1846/50
Adapt. 1 (5.7)	0.27/120/36.5	1.12/557/45	2.34/1101/**46.7**
Adapt. 2 (5.8)	**0.08/35.1/33.5**	**0.42/192/41.7**	**1.79/825**/47

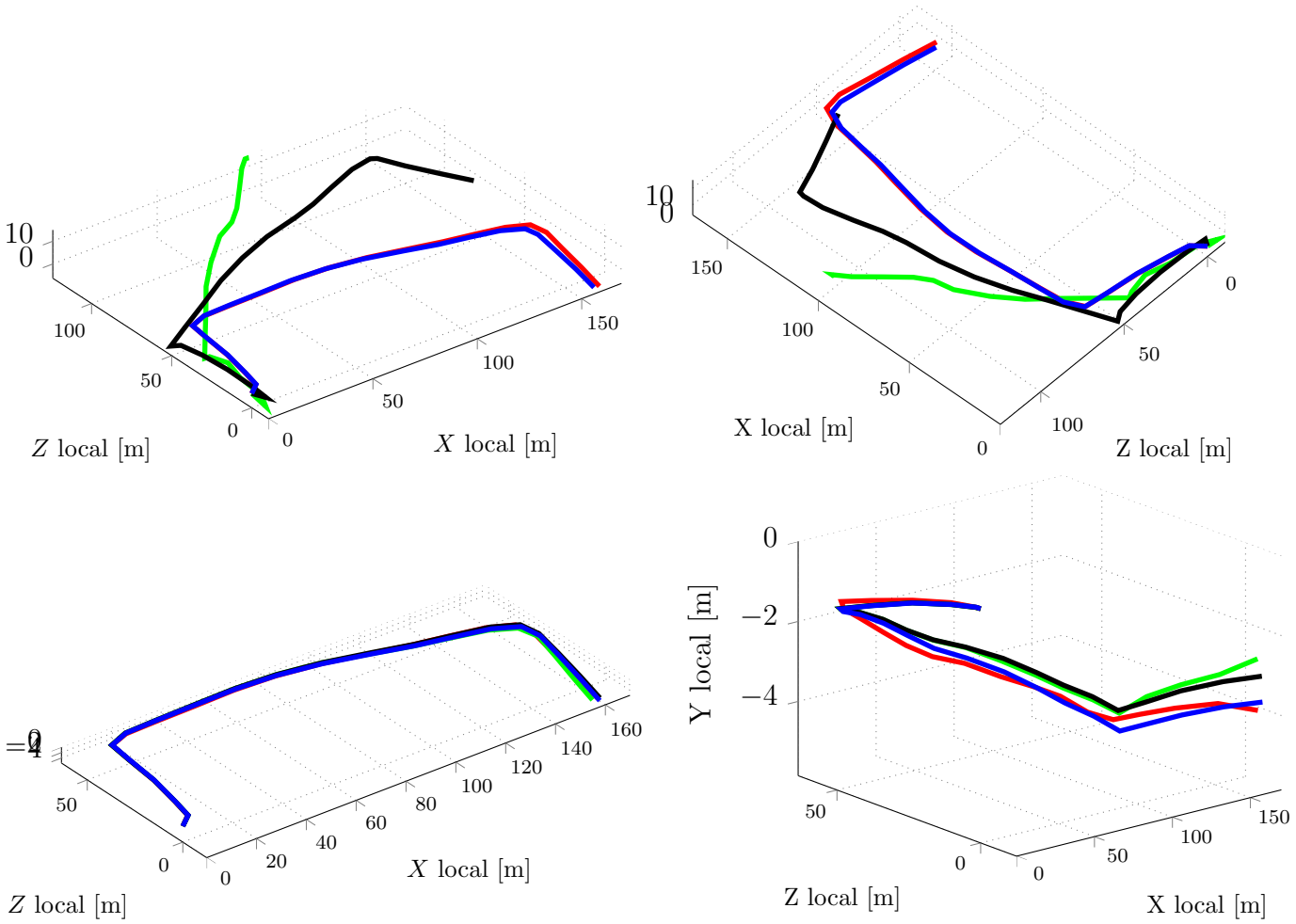

Figure 5.12 – Trajectory comparison for the RGB-D and adaptive formulations in the first 200 meters of the KITTI sequence 00, without (first row) and with multi-resolution (second row) with a gap of one frame. The registration considering the classical RGB-D is presented in green, the adaptive using (5.7) in black, (5.8) in blue and the ground truth trajectory in red. The adaptive registrations are notably more accurate in the fixed resolution. Notice the improvement of all techniques by using the multi-resolution (second row trajectories), where the discrepancy between the formulations is reduced. Although the accuracy of the methods is similar in this section of the trajectory, the adaptive formulations are still more accurate, as can be seen in the full trajectories shown in fig. 5.13 and on Table 5.4 for the gaps of 1, 2 or 3 frames.

of the approach in this context using the KITTI perspective sequence (see Table 5.4). The maximum number of iterations was of 50 at each pyramid level. To account the different computational cost of one iteration between the levels, we define the total number of iterations as $\sum_{i=1}^{4} l_i(2^{4-i})^2$, with l_i the number of iterations at level i. We can draw some remarks from the combination of the multi-resolution framework in the experiments. First, as expected, multi-resolution improved all the techniques (for instance, please see the upper and lower trajectories at fig. 5.12). Second, the discrepancy in the accuracy, observed between the methods in the fixed resolution, is reduced with multi-resolution. Still, the adaptive formulation remains more

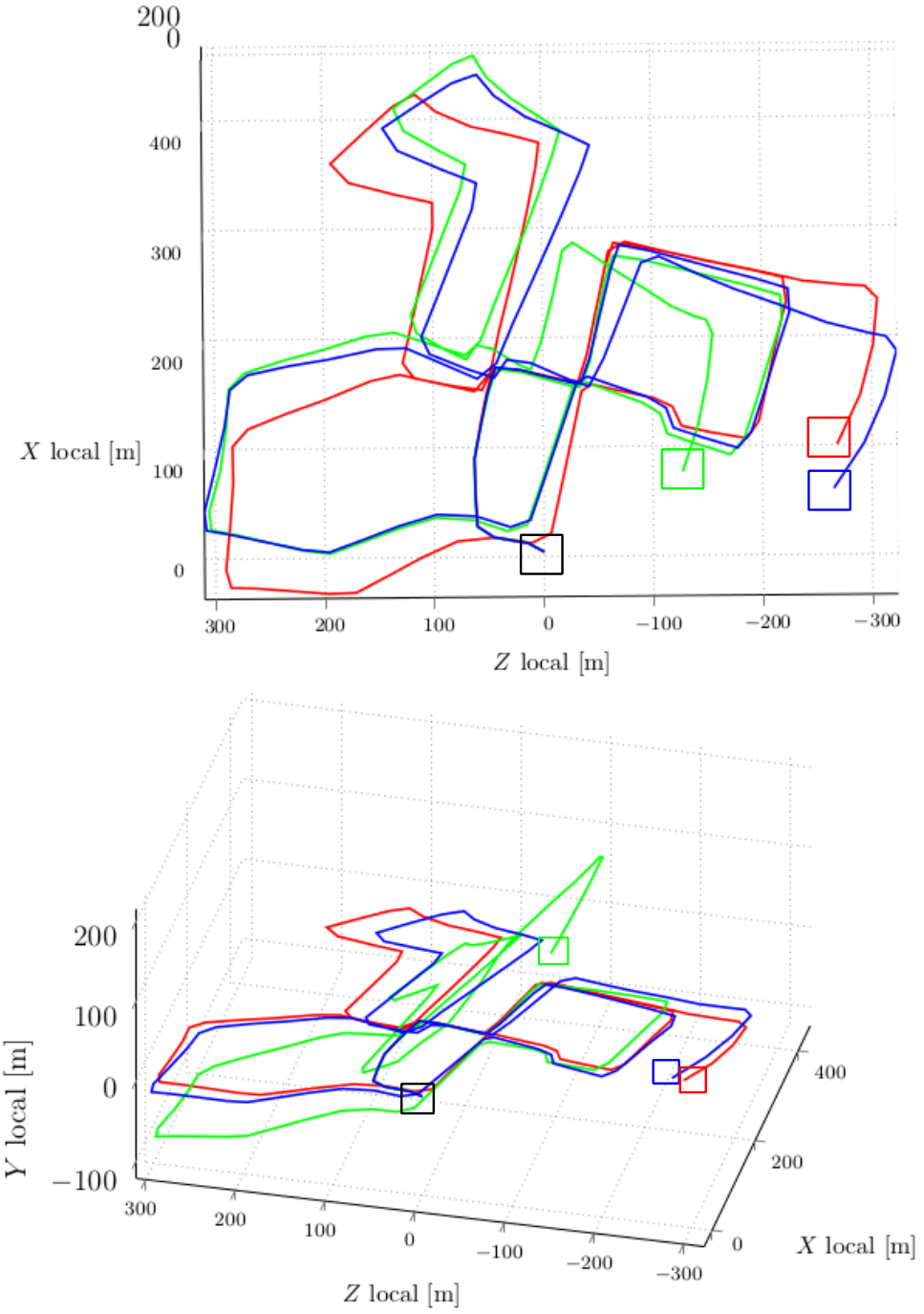

Figure 5.13 – Trajectory comparison for the classic RGB-D and adaptive formulations in the full KITTI sequence 00, both combined with multi-resolution with a gap of one frame. The upper figure is the top-view of the trajectory and the lower image is a lateral view. The registration considering the classical RGB-D is presented in green, the adaptive using (5.8) in blue and the ground truth trajectory in red. The start point is indicated by the black box and the endpoints of each trajectory by the boxes with respective colors. Both techniques present drift in the positioning in this long sequence (which is expected because no bundle adjustment or loop closure is performed), although the accumulated errors are smaller in the adaptive one.

Table 5.4 – Quantitative results using the KITTI outdoor sequence with **multi-resolution** (pyramid of four levels): average RRE[deg]/RTE[mm]/iterations.

	Gap = 1	*Gap = 2*	*Gap = 3*
Meth. 1 [Tykkala et al., 2011]	0.08/23.1/**447**	0.78/268/980	3.68/1059/1872
Adapt. 1 (5.7)	**0.06**/**16.5**/704	**0.19**/81.4/**856**	**0.83**/251/**1078**
Adapt. 2 (5.8)	**0.06**/**16.4**/1102	0.37/**47.5**/1269	1.05/**238**/1473

efficient and accurate, as can be observed in Table 5.4 and in the full trajectory of the KITTI sequence 00 shown in fig. 5.13.

5.5 Conclusions and Closing Remarks

This chapter proposed an RGB-D registration approach in the context of large inter-frame motions. The technique exploits adaptively the photometric and geometric error terms based on their convergence characteristics. Despite its simplicity, this technique was capable of dealing with large motions, occlusions and moving objects in indoor and outdoor real sequences. The proposed strategy improved the registration with simulated and real sequences, using both perspective and spherical sensor models.

We remark that it would be pertinent to further analyze theoretical properties of the photometric and geometric costs. The design of the scaling strategies were supported mainly by empirical observation and by the level curves shapes, but without any convergence proof. Notably, this line of research is correlated to the problem of defining the convergence domain of the different cost functions, i.e., the definition of upper bounds of convergence for direct image registration, in analogy to the "learned model" established in [Churchill et al., 2015, Dequaire et al., 2016] for feature-based registration techniques. In this context, some future directions include: *i)* the formal characterization of the convergence domain for different symmetries and noise statistics for both intensity and geometry costs; *ii)* the design of the costs with additional dense/semi-dense information such as planes, lines and image moments; and *iii)* combining this formulation with automatic local bundle adjustment and loop closure (RGB-D SLAM) for producing large-scale drift-free trajectories.

Part III

RGB-D Compact Mapping for Direct Image Registration

Introduction

In the following chapters, we will present strategies to reduce frame noise and to build compact photometric and geometric models of the environment. The depth in the spherical stereo frames is particularly noisy, often leading to local minima in RGB-D registration. This is noticed specially in the translation estimation. Conversely to indoor frames, smoothing and downsampling operations are not sufficient to reduce the noise in the outdoor frames. An illustrative example is given in fig. 5.14 with rendered views of the raw point cloud (left image) and the smoothed and downsampled point cloud using a Gaussian pyramid of four levels (center image). One can notice the large oscillations in the road and in the buildings facades. Unfortunately, the smoothed point cloud still contains artifacts in these regions. In order to reduce this noise, chapter 6 presents a segmentation based on superpixels using simultaneously the surface appearance (color) and orientation, encoded by the surface normals. This segmentation aims to drive the depth regularization in order to reduce the noise whilst maintaining semantically meaningful surfaces. This regularization increases the accuracy and reduces the number of iterations for convergence of the registration. We also note that each frame can be represented by a reduced number of planar patches after regularization.

Once the frames are regularized and their relative position estimated, we can build a compact photometric and geometric model describing the environment. Our compact mapping framework is based on keyframes distributed in the scene forming a topological-metric representation. This is specially relevant to reduce storage issues caused by redundant frames in the model. Different strategies can be adopted to the positioning and selection of keyframes, such as the photometric/geometric appearance changes or spatial distance between successive frames. In chapter 7, we describe a framework using the free space of the environment in order to build a compact keyframe-based map with good visibility and coverage properties.

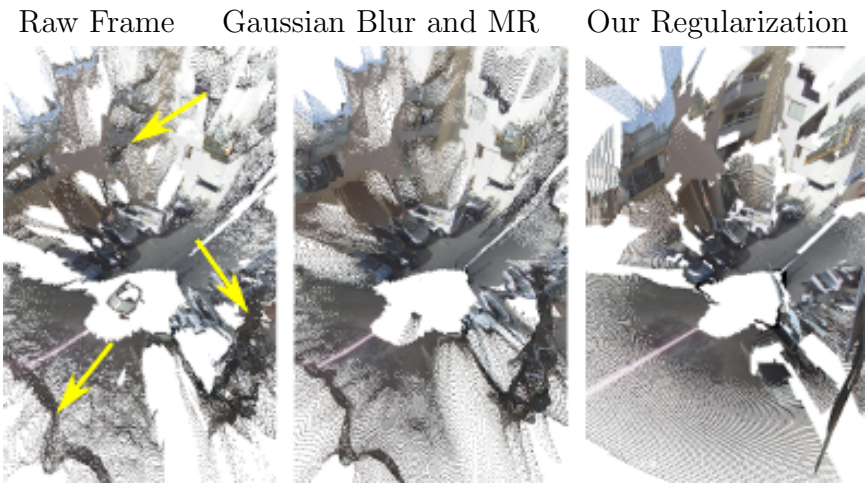

Figure 5.14 – Rendered views of the raw point cloud (Raw Frame), smoothed and with multiresolution (Gaussian Blur and MR) and using the regularization proposed in chapter 6. Some noisy and deformed regions are indicted by the arrows in the first image.

Chapter 6

Frame Regularization in Piecewise Planar Patches

Contents

6.1 Introduction

Despite being a classical problem of early vision and photogrammetry, estimating dense and accurate depth information from images remains a challenging and active research area. This is a difficult problem because of many factors including the occlusions between the views, appearance changes, textureless regions, repetitive patterns, sensor noise and the discontinuities of the objects boundaries in the scene. Furthermore, the depth estimation is also conditioned to the movement of the camera. Having a good estimate of the depth of the scene affects applications in diverse areas, such as architecture, archaeology, virtual and augmented reality and in robotics as, for instance, in registration and mapping techniques. We remark that these diverse applications have different requirements for the depth completeness and accuracy. For instance, a depth image containing over-regularized objects boundaries or with missing depth values can be acceptable for RGB-D registration or for compact mapping from planes (e.g., [Fernandez-Moral et al., 2013, Henry et al., 2014]). On the other hand, representing the details of objects boundaries can be expected in realistic scene rendering. Therefore, the expected accuracy and completeness of the depth are closely related to the specific user application.

In this chapter, we describe a framework to reduce the noise affecting the depth from stereo in the RGB-D frames, for registration and compact mapping applications. We remark that unlike the indoor RGB-D frames, smoothing and downsampling operations are not sufficient to reduce the noise in the outdoor depth images. The main objective is to build more accurate depth images by considering the scene composed of piecewise planar patches. This formulation is based on region growing approaches and it is adapted to the non-uniform resolution of wide FOV images. In summary, we exploit the complementarity of color and surface orientation to reduce noise and the uncertainty. These operations increase the accuracy of the depth (e.g., by reducing the noise from wrong pixel block matching), reduce the scene complexity and simultaneously enforces surface regularization in smooth regions. Furthermore, we extend the original RGB-D frames with an additional layer of information which encodes the uncertainty (a confidence layer C) of each pixel, i.e., the regularized frames have RGB-D-C layers ranking the points uncertainty.

The reminder of this chapter is organized as follows. Section 6.2 presents some related segmentation and regularization approaches. In section 6.3, we discuss the characteristics related to depth from stereo using panoramic images. Section 6.4 describes the photo-geometric surface segmentation and regularization. The creation of an uncertainty layer is discussed in section 6.4.4. In section 6.5, we present the processing of simulated and real frames of outdoor stereo sequences, showing that both accuracy of the depth and the appearance of the 3D scene model can be greatly improved. Finally, we summarize the chapter in section 6.6.

6.2 Related Works

For creating more accurate depth maps from multi-view frames, [Huhle et al., 2010, Zhang et al., 2012, Vogel et al., 2013] proposed energy based regularization techniques exploring complementary aspects such as surface smoothness and temporal prediction to explicitly reduce the effects of surface discontinuities and occlusions. [Schönbein and Geiger, 2014] computed the depth image from ominidirectional stereo images by complementing SGBM [Hirschmuller, 2008] with a regularization driven by the hypothesis of a Manhattan World with large dominant planes. Their approach explored also the temporal constraints of subsequent stereo frames by performing a fusion of the two closest neighboring frames for filling gaps at the reference frame. Similarly, [Wang et al., 2016] proposed an incremental and enhanced scanline-based segmentation method to augment 3D lidar point clouds into planar super-resolution patches. The goal of their work is to reduce the rate of over and miss-segmentation induced by the sparsity and non-uniformity of Velodyne point clouds. Both aspects are considered in our regularization.

Edge-preserving filters combined with partial differential equations (PDE) are very popular in the computer vision community, and they are used in many applications such as noise reduction, stereo matching, image deconvolution and image upsampling (e.g., [Perona and Malik, 1990, Chambolle, 2004, Chambolle and Pock, 2011, Newcombe et al., 2011b]). For instance, [Perona and Malik, 1990] proposed the anisotropic diffusion filter to regularize images while preserving region boundaries. Within a similar goal of preserving sharp boundaries, [Chambolle, 2004, Chambolle and Pock, 2011] described a general total variation (TV) framework and practical implementation algorithms to many image denoising applications. The main drawback of these techniques is their large computational burden. For illustration, consider the simple case of denoising a 1D signal of length 180, as shown in fig. 6.1. This signal is corrupted with a Gaussian noise of mean zero and standard deviation of four. We consider the Rudin-Osher-Fatemi (ROF) total variation model and the algorithm to optimize this energy is described in section 6.2.1 of [Chambolle and Pock, 2011], resulting in a TV gradient kernel of size 180×180. The weighing factor for the data term was chosen as $\lambda = 0.01$ and the denoised signal after 400 iterations is shown in the right plot. In general, for a 2D image of dimensions $m \times n$, the number of variables is $N = mn$ and these kernels are $N \times N$, which is a relatively large problem even for low resolution images used in chapters 4 and 5. The regularization technique presented in this chapter is simpler, more efficient and reduces the noise of dominant surfaces in the scene, but it is less accurate in object borders. However, the regularized depth has enough accuracy for image registration and compact mapping applications. In order to reduce the size and complexity of the optimization, recent methods explored a prior segmentation or "simplification" [1] of surfaces and 3D point clouds (e.g., [Edelsbrunner, 2000, Shamir,

1. Surface simplification is the process of reducing the number of parameters used to describe a surface while keeping the overall shape and boundaries preserved.

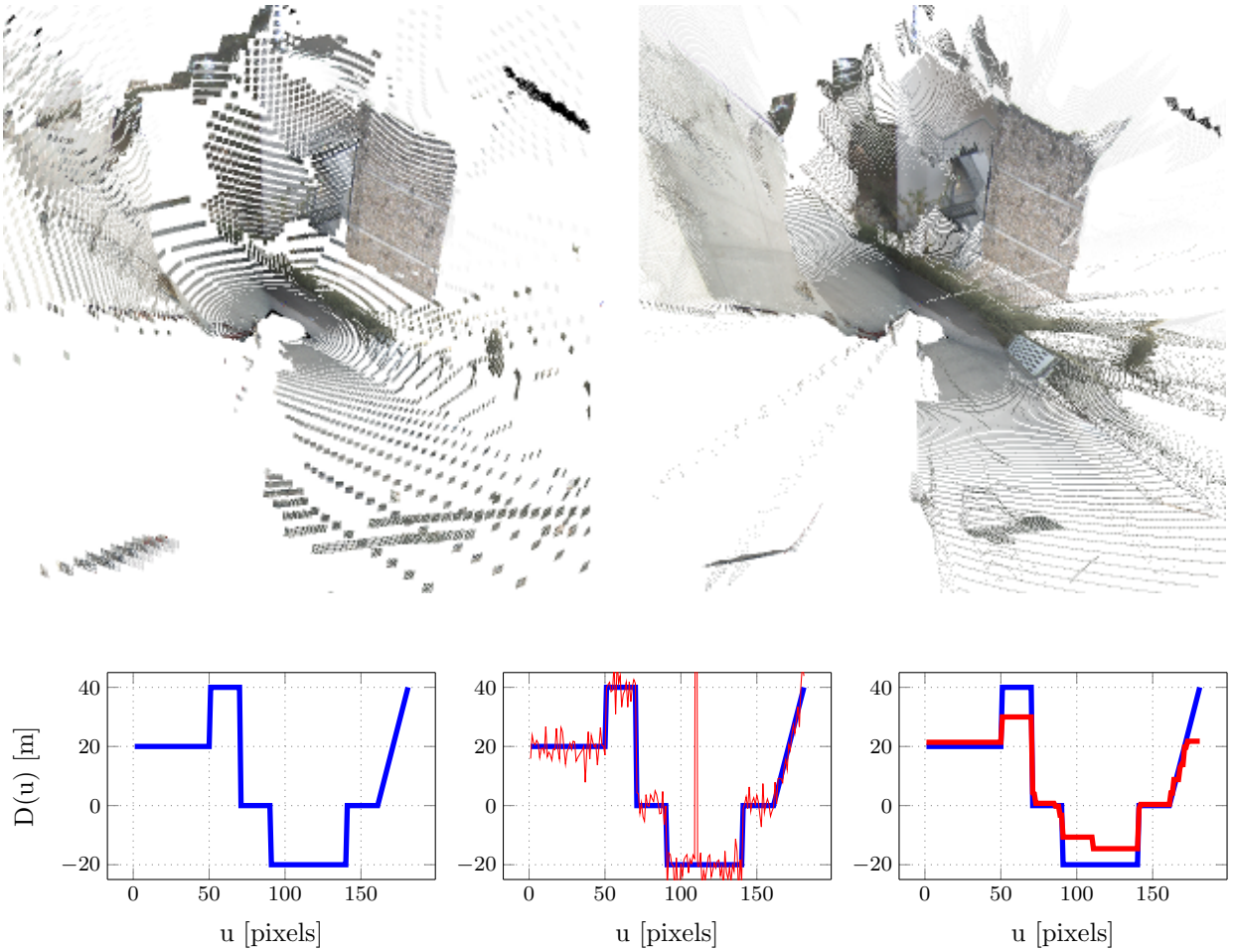

Figure 6.1 – Interpolation of the depth and regularization of a unidimensional signal using total variation. The top row depicts two upsampled point clouds of an outdoor frame with the nearest neighbour (at left) and bilinear (at right). As can be noticed the interpolation method can change considerably the consistency of the final signal. In the bottom row, a 1D signal (at left) is corrupted with Gaussian noise of zero mean and standard deviation of four (center figure in red). The denoised signal using total variation regularization with the ROF model is displayed in the plot at right (in red). Notice the staircasing effect in the affine part of the signal.

2008, Schönbein and Geiger, 2014, Duan and Lafarge, 2015, Wang et al., 2016]). One can generate sub-models partitioning the input domain into elementary cells, then reduce the number of degrees of freedom and explore constraints of neighboring regions. In this context, superpixels based on region growing have been recently used for disparity computation [Vogel et al., 2013, Yamaguchi et al., 2014, Schönbein and Geiger, 2014] and segmentation of RGB-D images [Weikersdorfer et al., 2013]. Inspired by these works, the proposed regularization approach uses an intermediary superpixel segmentation. We extend the Simple Linear Iterative Clustering (SLIC) [Achanta et al., 2012] to RGB-D panoramic images.

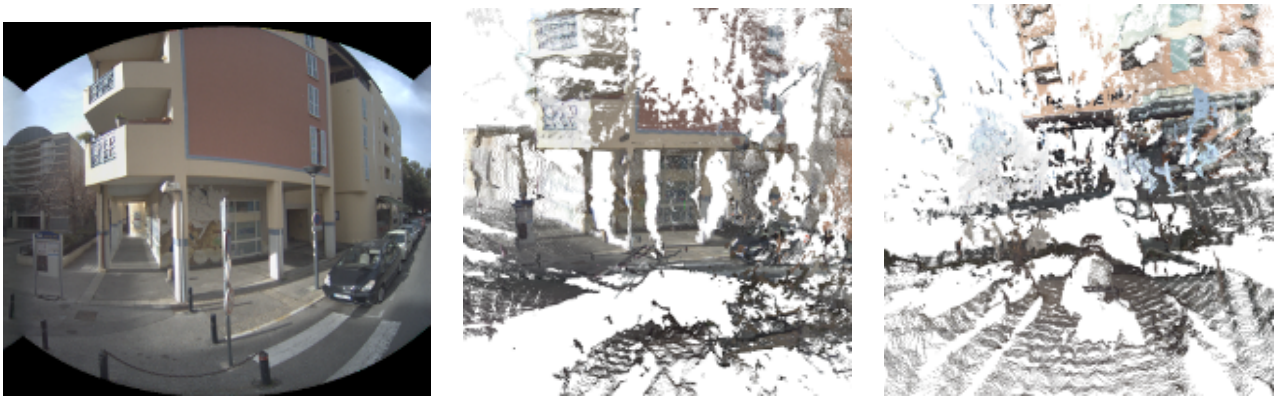

Figure 6.2 – Lateral and top views of a point cloud using SGBM stereo. The left image depicts a region of interest of an equirectangular spherical frame, followed by the lateral and top views of the reconstructed point cloud. We remark the oscillations present at planar surfaces such as the floor and at the facade of the building.

6.2.1 Contributions

In this chapter, we propose an extension of the state-of-the-art SLIC superpixel algorithm to segment RGB-D images including geometric information beyond color, such as the surface normals encoded as a color image. We also present an adapted version of SLIC to omnidirectional images by using the geodesic in the sphere, instead of the Euclidean norm.

6.3 Background and Spherical Stereo

In this chapter, the frames are composed of color and depth images: $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}\}$. The final regularized frame has an additional uncertainty/confidence layer $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{C}}\}$. Finally, a planar patch Γ_i is represented by the pair $(\mathbf{n}_i, d_i)$, such as for any 3D point $\mathbf{P} \in \Gamma_i$: $\mathbf{n}_i^T\mathbf{P}+d_i = 0$. In the following sections, we introduce the segmentation and posterior regularization.

6.3.1 Characteristics of Depth from Spherical Stereo

The depth images of each frame can be acquired using active sensors (e.g., LIDAR, RGB-D cameras) or from stereo, as introduced in section 2.3. Considering the stereo case, the depth errors come mainly from three different sources: *i)* wrong pixel matching assignments, particularly in low textured surfaces; *ii)* occlusions and pixels without co-visibility; and *iii)* violation of the brightness constancy assumption coming from reflections, mirrors and translucent structures. Besides, the resulting depth image characteristics are closely related to the selected stereo matching technique. For instance, SGBM [Hirschmuller, 2008] gives less smoothed depth images and with more missing depth values than ELAS [Geiger et al., 2010]. On the other hand, ELAS computes more complete and smoothed disparity images which are, besides being as accurate as SGBM, "pleasant" images. However, it also produces border blending artifacts,

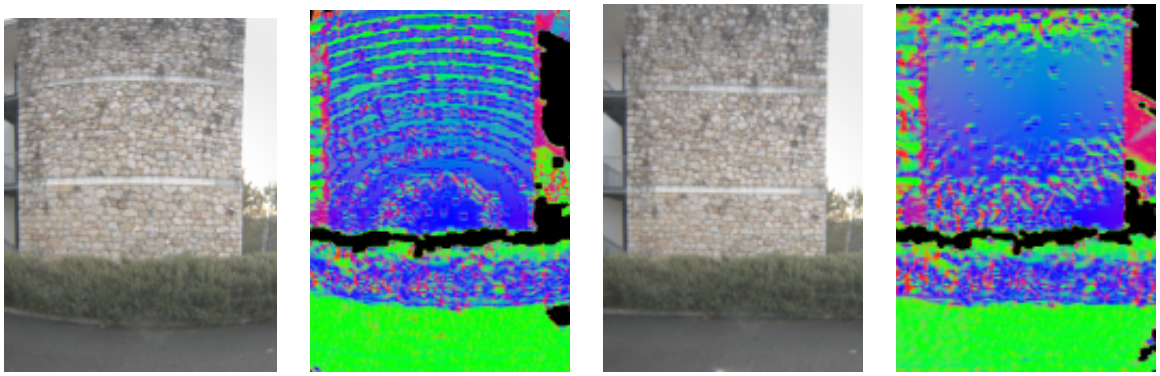

Figure 6.3 – Normal vectors of a planar region using ELAS stereo in the sphere (two first columns) and using ELAS stereo with perspective projection (two last images). The oscillation component induced by the regularization and the spherical spatial non-uniformity is clearly depicted in the spherical image.

where posterior filtering might have marginal influence. Similar conclusions were observed in [Wang et al., 2014].

Some additional difficulties arise when performing stereo of omnidirectional images. For instance, most disparity computation methods are developed for perspective images, i.e., using the Euclidean metric with convex energy minimization techniques that flavor piecewise linear solutions (e.g., the TV ROF regularization as shown in fig. 6.1). This phenomenon is known as staircasing effect of affine signals [Muller et al., 2011, Pinies et al., 2015] and it can be noticed in the denoised signal of fig. 6.1. Furthermore, the current state-of-the-art disparity algorithms (e.g., SGBM [Hirschmuller, 2008] and ELAS [Geiger et al., 2010]) assume priors to assign disparity values to less informative image regions (see fig. 6.3). This constraint is not drastic with perspective images because most surfaces can be modeled as locally planar in the Euclidean domain. Furthermore, scenes in human-like environments can be often locally approximated by planar patches, which remains a plane in perspective stereo. Unfortunately, even this local assumption is not valid in the sphere because a plane has no finite polynomial expansion, and the linear disparity solutions of these algorithms induce oscillations in the sphere. The computation of disparity in the sphere, therefore, greatly suffers from this effect.

Possible strategies to reduce these shortcomings are to redefine the perspective image processing and optimization tools in the manifold; or to unwarp the spherical image into a planar perspective view (e.g., using two or more virtual perspective cameras, such as the cube projection using four perspective cameras, as shown in fig. 6.3). However, this strategy presents two drawbacks. The first concerns its inefficiency, in terms of both computational cost and memory usage. Secondly, the unwarping of the non-uniform resolution of the original image is not taken into account and can generate artificial artifacts in the unwarped view. Another possibility is to regularize the resulting depth image as discussed in the next sections.

6.4 Frame Regularization in Piecewise Planar Patches

In this section, we describe region growing techniques to segment the depth image into piecewise planar patches. The depth image is then represented as a set of planes of variable size, where non-planar surfaces are approximated by a set of small planes. This assumption is applicable for any environment: structured and non-structured as long as the planar approximation error is smaller than the measurement error. This can be achieved by selecting the suitable region growing parameters according to the sensor noise model, regardless the type of scene. We present two region growing strategies. The first is a geometric region growing segmentation in the Euclidean 3D space. Subsequently, we propose a region growing formulation exploring both color and geometry in the sphere. This second formulation is an adapted version of SLIC superpixels to ominidirectional RGB or RGB-D images (OmniSLIC).

6.4.1 Geometric Region Growing in Euclidean 3D Space

A first approach to segment the depth is to consider a purely geometric region growing algorithm. The segmentation is then based on growing regions around "seeds" that are distributed along the depth image, e.g., using k-means. The neighboring points to each seed are then tested following two metrics and if these two criteria are fulfilled within suitable thresholds, then the 3D point is included in the growing patch. We present in a first moment a purely geometric region growing, with the following conditions. The conditions to insert a 3D point at pixel $\mathbf{p}$ in a candidate patch are: *(i)* if the normal vectors of a pixel $\mathbf{p}$ and the planar patch have similar direction; and *(ii)* if the orthogonal projection error is small, i.e., the 3D point corresponding to the pixel $\mathbf{p}$ lies approximately on the patch. These patches grow until a stop criteria, usually based on the limits of the neighborhood or the maximum number of points allowed to avoid aliasing. Since the surface resolution decreases with the distance to the sensor, an adaptive number of allowed points at each patch is employed to build isometric patches (in the 3D space). This avoids undesirable effects as aliasing details of far objects and the oversampling of structures close to the sensor. The number of allowed points per patch depends on the desired area A and the spatial density point's distribution in the sphere of radius $\boldsymbol{\mathcal{D}}(\mathbf{p})$: $\mu(\boldsymbol{\mathcal{D}}(\mathbf{p})) = N/(4\pi\boldsymbol{\mathcal{D}}(\mathbf{p})^2)$, with N being the total number of pixels in the depth image. Given the number of points $n_1 = \mu(\boldsymbol{\mathcal{D}}(\mathbf{p}_1))A$ at range $\boldsymbol{\mathcal{D}}(\mathbf{p}_1)$, a same area patch at range $\boldsymbol{\mathcal{D}}(\mathbf{p}_2)$ has a number of points:

$$n_2 = n_1 \left(\frac{\boldsymbol{\mathcal{D}}(\mathbf{p}_1)}{\boldsymbol{\mathcal{D}}(\mathbf{p}_2)} \right)^2 . \tag{6.1}$$

Then, it is sufficient to consider an interval around the value of n_2 as minimum and max points in the region growing procedure, as presented in the algorithm 6.4.1.1. This scheme reduces

scene complexity and simultaneously applies surface regularization to reduce the oscillations. Furthermore, this isotropic segmentation avoids undesirable effects such as of aliasing details of distant surfaces and the over-sampling of surfaces near to the camera. However, this algorithm alone is often not capable of reducing the noise from the spherical stereo rig. An example of such regularization is given in fig. 6.4 for two frames. In order to combine coherent neighboring patches, we proposed in [Martins et al., 2015] to use also the color information from superpixels to increase the size of the patches. The photometric superpixel then would limit the size of the regularized patch. A more suitable policy would consist of using color information simultaneously to geometric information to aggregate neighboring patches, as described in the next section.

6.4.2 OmniSLIC: Omnidirectional SLIC Superpixel

In this section, we describe an adaption of the state-of-the-art SLIC superpixel segmentation [Achanta et al., 2012] to panoramic RGB and RGB-D images. The reason for selecting SLIC is threefold. First, SLIC has nice properties as strong adherence to boundaries and compactness. Second, it is of easy tuning because the method has two parameters: the expected maximum number of superpixels (k) and a parameter relating the compactness of the superpixel to border adherence (m). Finally, it has a simple implementation similar to the region growing algorithm described in the previous section 6.4.1, while being more efficient. We give in the following a brief summary of the basic principles used in the segmentation as described in [Achanta et al., 2012]. SLIC computes color and spatial distances of the pixels to a candidate cluster, whose mean color and pixel position are ($\boldsymbol{\mathcal{I}}_s(\bar{\mathbf{p}})$,$\bar{\mathbf{p}}$):

$$\boldsymbol{\Upsilon}(\mathbf{p}) = \sqrt{\boldsymbol{\Upsilon}_c^2(\mathbf{p}) + \left(\frac{\boldsymbol{\Upsilon}_s(\mathbf{p})}{S}\right)^2 m^2} \tag{6.2}$$

Algorithm 6.4.1.1 : Geometric segmentation in the Euclidean 3D space

Input 1: reference area and distance (ρ_1 and A): $n_1 = \mu(\rho_1)A$
Input 2: max errors ε_n and ε_d
Distribute seed planes in the pixel positions $\bar{\mathbf{p}}_\mathbf{1}, \bar{\mathbf{p}}_\mathbf{2}, ..., \bar{\mathbf{p}}_\mathbf{n}$
for all $\bar{\mathbf{p}}$: compute max and min points $n_2 = n_1 \left(\frac{\rho_1}{\boldsymbol{\mathcal{D}}(\bar{\mathbf{p}})}\right)^2$ **do**
  for all pixels $\mathbf{p}$ neighboring $\bar{\mathbf{p}}$ **do**
    $n_{min} = \max(n_2 - 5, 5)$ and $n_{max} = n_2$
    while $size(\Gamma) < n_{max}$ & $\|\mathbf{n}^T\mathbf{n}(\mathbf{p})\|_1 < \varepsilon_n$ & $\|\mathbf{n}^T\mathbf{P}(\mathbf{p}) + d\|_1 < \varepsilon_d$ **do**
      Re-evaluate plane patch Γ including $\mathbf{P}(\mathbf{p})$
    end while
  end for
end for

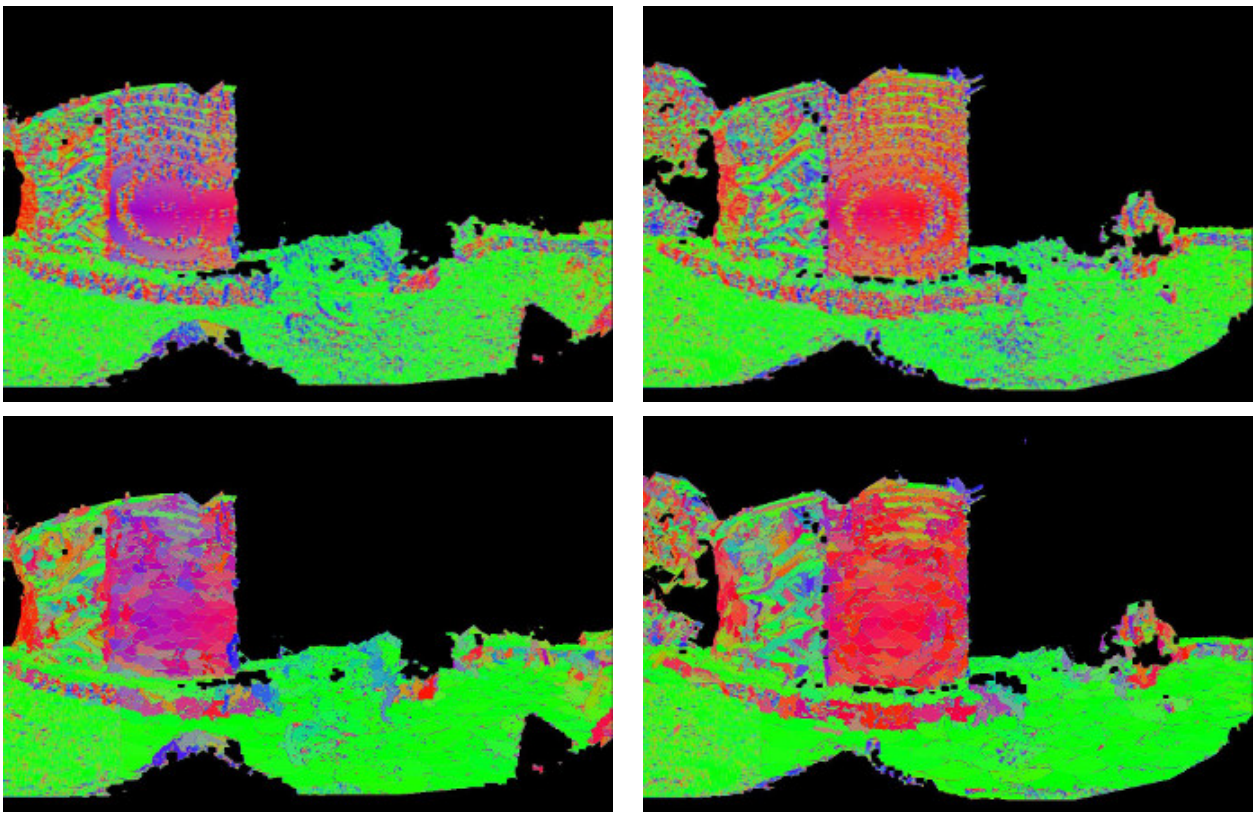

Figure 6.4 – Examples of regularization using geometric patches. The upper row depicts the normals with the original depth images. The lower row shows the correspondent images after the regularization.

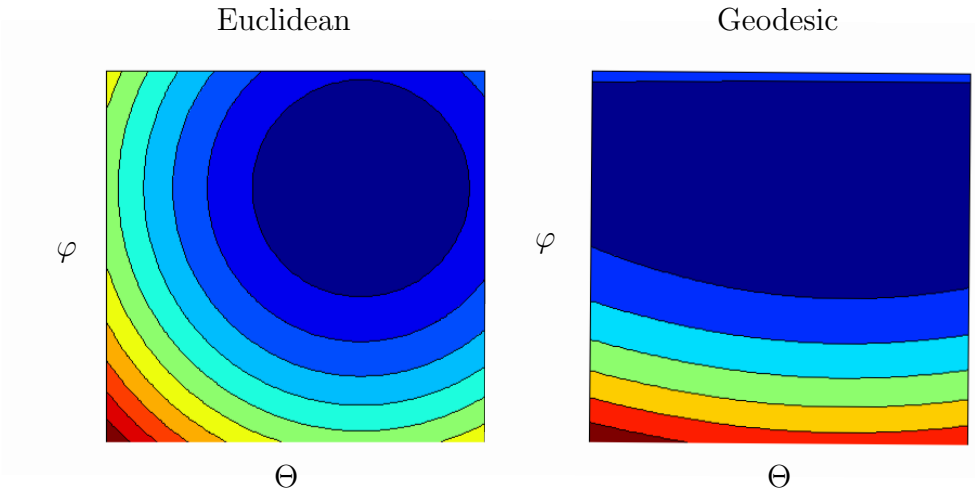

Figure 6.5 – Euclidean and geodesic distances for pixels in the top right corner of a spherical image. Observe that the geodesic distance increases mainly in the φ direction, allowing superpixels to have wider widths in the Θ direction, in this part of the sphere.

with the color images encoded in the CIELAB space. The color image is projected in this color space because CIELAB color difference is more perceptually uniform than in other color spaces,

such as RGB or HSV. The respective color and spatial distances are given by:

$$\boldsymbol{\Upsilon}_c(\mathbf{p}) = \|\boldsymbol{\mathcal{I}}(\mathbf{p}) - \boldsymbol{\mathcal{I}}_s(\bar{\mathbf{p}})\|_2 \quad \text{and} \quad \boldsymbol{\Upsilon}_s(\mathbf{p}) = \|\mathbf{p} - \bar{\mathbf{p}}\|_2. \tag{6.3}$$

The tuning parameters of SLIC in (6.2) are S and m which act as normalization factors of the color and spatial distances. The parameter m is given by the user (is the compactness factor) and S depends on the expected number of superpixels and their shape, e.g., squared, hexagonal or circular. For instance, $S = \sqrt{N/k}$ for a square shape or $S = \sqrt{2N/\sqrt{3}k}$ for a hexagonal one.

6.4.2.1 Omnidirectional RGB-D SLIC

The modifications included in the original SLIC algorithm are twofold. First, we propose to account the non-uniform resolution of omnidirectional images for the spatial distance term. Second, the modified SLIC can consider simultaneously color, geometric constraints such as depth and the normals for defining the superpixels. The tradeoff between the intensity and geometric constraints are set by selecting one tuning parameter.

The first modification is to consider the spatial distance metric in e.q. (6.3) as the geodesic in the unit sphere $\mathbb{S}^2$:

$$\boldsymbol{\Upsilon}_s(\mathbf{p}) = \arccos(\Pi_S^{-1}(\bar{\mathbf{p}})^T \Pi_S^{-1}(\mathbf{p})). \tag{6.4}$$

The shape of the superpixels in wide FOV images can be better conditioned using the spatial distance in the sphere as shown in fig. 6.5. Observe that the spatial distance increases mainly in the φ direction in the geodesic, allowing superpixels to have wider lengths in the Θ direction. An example for comparison of the different superpixels using the Euclidean and the geodesic is given in fig. 6.6 using a fisheye image from the dataset in [Zhang et al., 2016]. The input parameters are $k = 50$ and $m = 30$ in both cases and the shape of some superpixels using the geodesic are more coherent than using the Euclidean distance. Some examples are indicated by green boxes in the original segmented fisheye image, as well as in the equirectangular warped image.

RGB-D Superpixels

For segmenting RGB-D images, we can explore geometric constraints beyond colors such as the normals. Interestingly, the normals can be represented as a color image ($\boldsymbol{\mathcal{N}}$):

$$\boldsymbol{\mathcal{N}}(\mathbf{p}) = \mathbf{n}(\mathbf{p}) \tag{6.5}$$

which can be treated similarly to color RGB image after its encoding in the CIELAB color space. The adapted distance using the distance in (6.2) for an RGB-D ominidirectional image

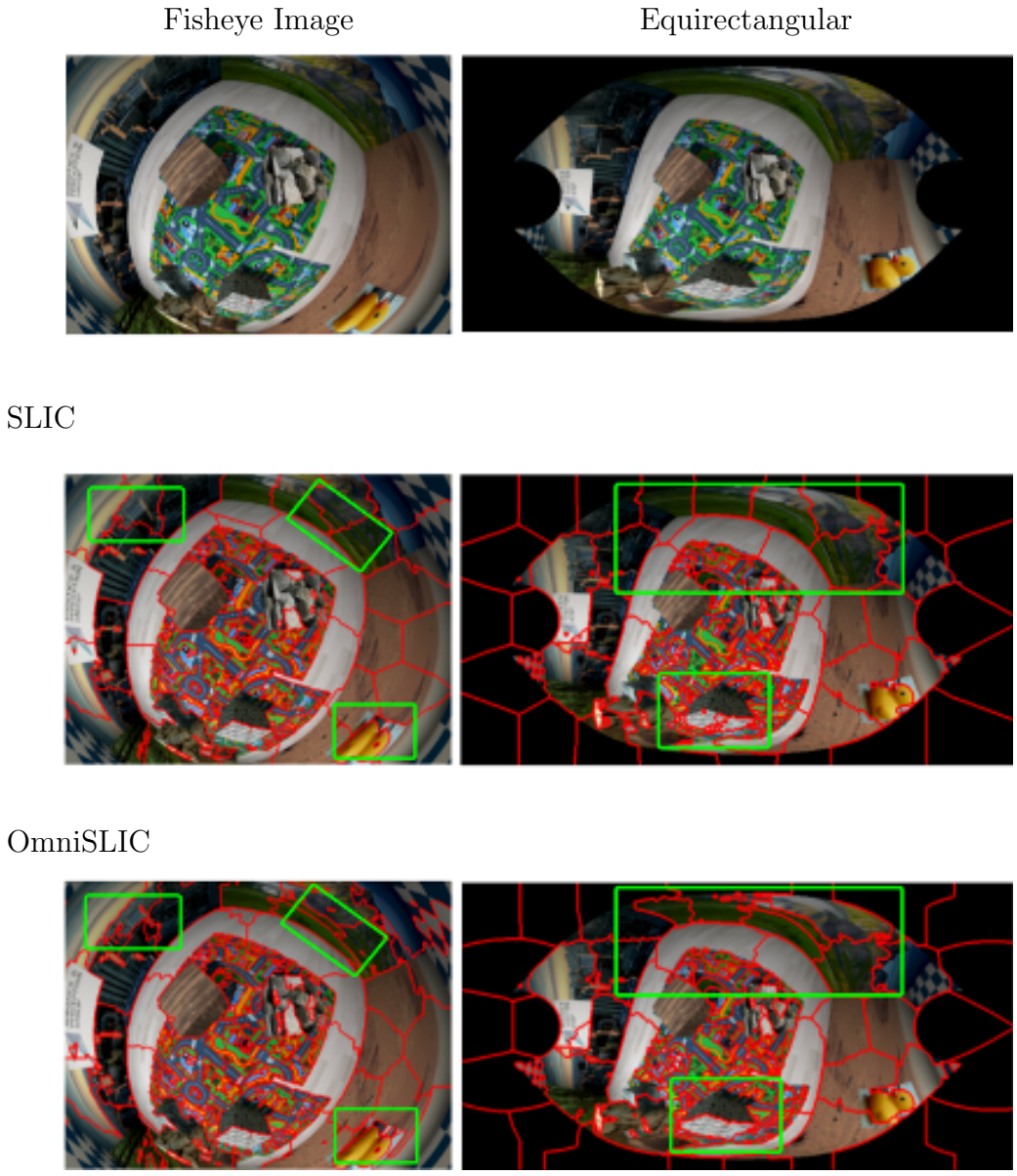

Figure 6.6 – SLIC and OmniSLIC superpixel segmentations of a fisheye image. The shape of some superpixels using the geodesic are more coherent than using the Euclidean distance. Some examples are indicated by green boxes in the original segmented fisheye image, as well as in the equirectangular image.

is then $\boldsymbol{\Upsilon}_o = \sqrt{(\boldsymbol{\Upsilon}|_{\mathcal{I}})^2 + (\lambda\boldsymbol{\Upsilon}|_{\mathcal{N}})^2}$, that is:

$$\boldsymbol{\Upsilon}_o(\mathbf{p}) = \sqrt{\Upsilon_c^2(\mathbf{p}) + (1+\lambda^2)\left(\frac{\boldsymbol{\Upsilon}_s(\mathbf{p})}{S}\right)^2 m^2 + \boldsymbol{\Upsilon}_n^2(\mathbf{p})\lambda^2} \tag{6.6}$$

where the RGB distance term $\mathbf{D}_c$ is as in (6.3), the distance from the image of normals is as:

$$\boldsymbol{\Upsilon}_n(\mathbf{p}) = \|\boldsymbol{\mathcal{N}}(\mathbf{p}) - \boldsymbol{\mathcal{N}}(\bar{\mathbf{p}})\|_2 \tag{6.7}$$

and $\boldsymbol{\Upsilon}_s$ is using the distance in the sphere (6.4). The parameter λ makes the tradeoff between the boundaries of the RGB and the normal image. Setting $\lambda = 0$ is equivalent to define the superpixels only from color. A segmentation using both normals and color images is given in fig. 6.7 for a catadioptric image from [Zhang et al., 2016]. Observe that adherence to the boundaries of the superpixels takes into account the normal map (second row images) and the color images. The values used in the segmentation are $k = 50$, $m = 30$ and the tradeoff between the intensity and normal images is $\lambda = 1$. Finally, some examples using real spherical images are given in figs. 6.8, 6.9 and 6.10 for outdoor and indoor spherical RGB-D frames with values $\lambda = \{0, 1\}$.

6.4.3 Combining Adjacent Coherent Patches

A last stage is carried out to merge the neighboring superpixels that lie approximately in the same 3D plane and that have similar color. This is used to extract a skinny representation of the scene and to reduce the depth error in large planar regions: the ground floor, the facades and main planes. We used a customized version of the density based clustering (DBSCAN) segmentation algorithm [Ester et al., 1996] because it can exploit directly the output and the adjacency matrices of the SLIC superpixels [2]. The final segmentation is just the combination of adjacent superpixels with plane and mean color $(\Gamma_i, \boldsymbol{\mathcal{I}}_i)$ that are bellow a threshold from the computed mean patch superpixel $(\Gamma_s, \boldsymbol{\mathcal{I}}_s)$, i.e., for all neighboring superpixels i that:

$$\|\boldsymbol{\mathcal{I}}_i - \boldsymbol{\mathcal{I}}_s\|_2 < \varepsilon_1 \text{ and } \|\arccos(\mathbf{n}_i^T \mathbf{n}_s)\|_1 > \varepsilon_2.$$

Finally, a mean patch Γ_s can be extracted from each segmented region by using a robust plane fitting RANSAC algorithm [Hartley and Zisserman, 2003]. The set of pixels $\mathbf{p}$, such as depth $\boldsymbol{\mathcal{D}}(\mathbf{p})$ belongs to the segmented patch Γ_s, are then fulfilled by employing the resulting final

2. The basic implementation of SLIC and DBSCAN algorithms are from http://www.peterkovesi.com/matlabfns/index.html.

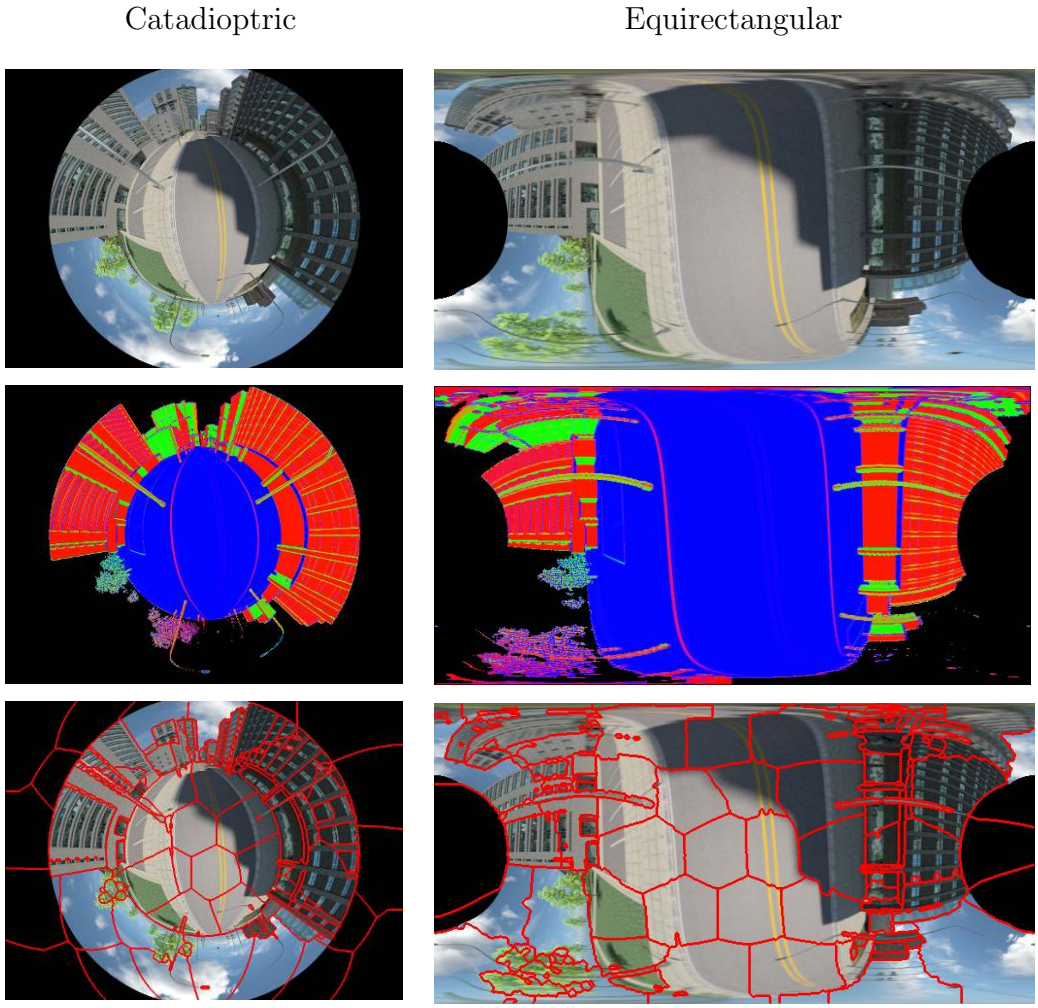

Figure 6.7 – OmniSLIC superpixel segmentations of an RGB-D catadioptric image. Observe that adherence to the boundaries of the superpixels takes into account the normal map (second row images) and the color images. The values used in the segmentation are $k = 50$, $m = 30$ and the tradeoff between the intensity and normal images is set to $\lambda = 1$.

patch parameters $(\mathbf{n}_s, d_s)$ as:

$$\boldsymbol{\mathcal{D}}_s(\mathbf{p}) = \left\| \frac{d_s}{(\mathbf{n}_s^T \Pi_S^{-1}(\mathbf{p}))} \right\|_1 . \tag{6.8}$$

6.4.4 Uncertainty Characterization

The errors coming from stereo are commonly supposed to be from disparity computation, which is related by the inverse of the depth in perspective stereo. Therefore, some works perform the uncertainty modeling considering the inverse depth in VSLAM or RGB-D tracking [Civera et al., 2008, Gutierrez-Gomez et al., 2016]. The inverse parametrization is interesting

for points far from the camera, which have a more stable uncertainty representation and that can constraint the rotation estimation. Due to the range displayed by the considered sensors in this thesis, we will represent the uncertainty of the depth as in most works, by propagating the disparity variance: $\sigma_D^2 \propto D^4$. The uncertainty characterization of the noise affecting depth images acquired from active sensors (LIDAR and Kinect) has received a broad attention from the robotics community [Khoshelham and Elberink, 2012, Dryanovski et al., 2013]. For instance, the Kinect v1 noise model can be described by $\sigma_D^2 = 2.05 \times 10^{-6} D^4$ for $D \in [0, 5]$ meters. In other words, the expected accuracy of this sensor is of up to 7cm for the maximum range with 95.4% confidence [Khoshelham and Elberink, 2012]. The assumption about the local planarity of the scene also helps to characterize this uncertainty because additional information of the patch neighborhood is provided. Thus, the uncertainty of the regularized depth measurement belonging to the patch can be modeled as:

$$\Sigma_{\mathcal{D}}(\mathbf{p}) = \frac{\boldsymbol{\mathcal{D}}_s^4(\mathbf{p})}{\|\mathbf{n}_s^T \Pi_S^{-1}(\mathbf{p})\|_1} \tag{6.9}$$

encoding the distance and the visibility of the point after the regularization, the denominator in (6.9) measures the point observability condition, i.e., the points in the patch, whose direction of view are more orthogonal to the normals ($\mathbf{n}_s^T \Pi_S^{-1}(\mathbf{p}) \approx 0$), should have higher uncertainty.

6.5 Experiments

Common quantitative metrics to evaluate segmentation/regularization algorithms are to compute the root-mean-square error (RMSE), peak signal-to-noise ratio (PSNR), border recall or the Jaccard index in the borders of objects. Such metrics require a precise 3D ground truth model of the scene and labelled class images of the different surfaces. Therefore, in this section, we perform a more qualitative analysis of the improvement of the frames by the segmentation/regularization.

In the experiments, we use low resolution images because the main goal is to produce depth to be used in compact mapping and registration. In this context, we do not have the requirement of maintaining sharp boundaries in the objects edges, but a more concise depth information with semantically meaningful boundaries, i.e., a depth segmentation and regularization of walls, facades, floor and other dominant surfaces. To produce higher resolution images, we perform upsampling with bilinear interpolation. However, it is worth noting that the region growing can be also applied directly in the highest image resolution to obtain less smooth frames for scene rendering and virtual reality immersion.

The tradeoff between the color and normal images (the scaling λ) depends on how structured is the scene and the level of noise affecting the normal images. For instance, if the scene contains vegetation or very unstructured surfaces, small values of λ should be considered in order to

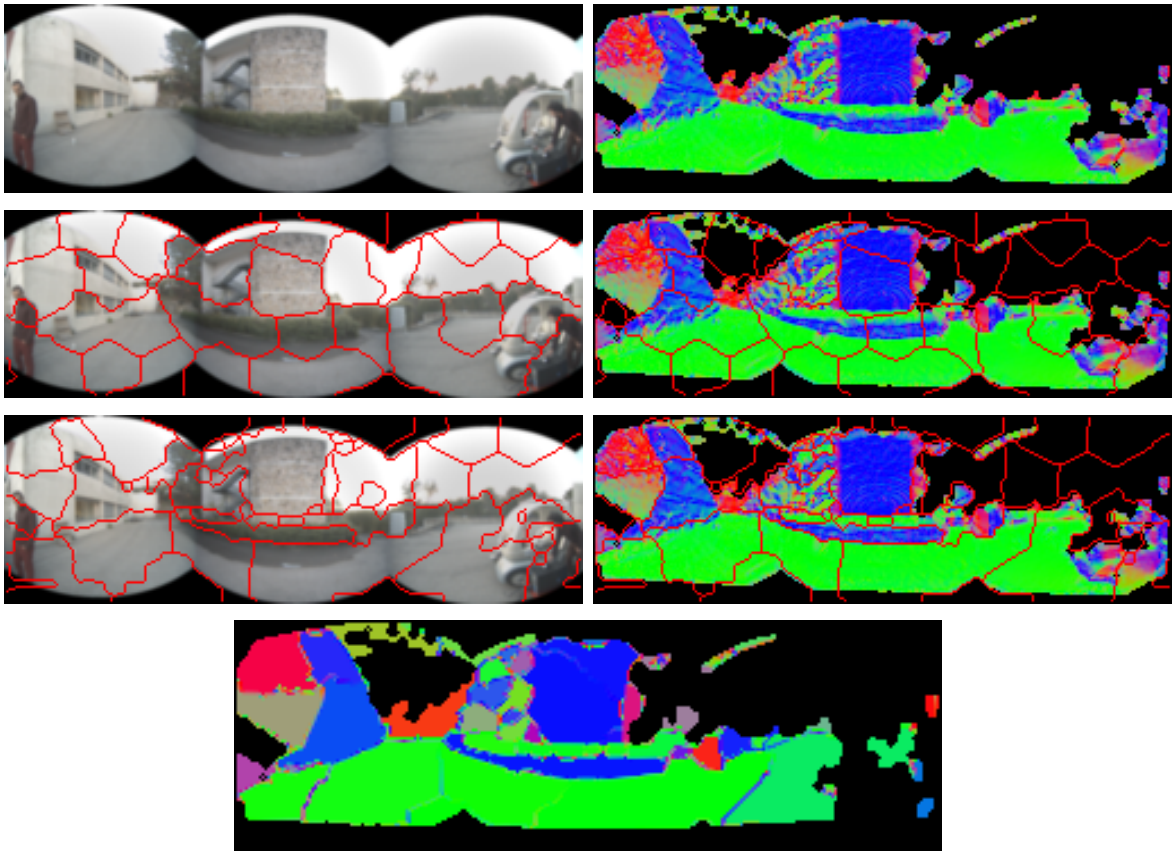

Figure 6.8 – OmniSLIC RGB-D image segmentation example for an outdoor frame. The first row depicts the RGB and normal images encoded by color. The second row is the resulting segmentation using $\lambda = 0$ (second row) or $\lambda = 1$ (third row). The normals of the regularized depth are shown in the bottom image.

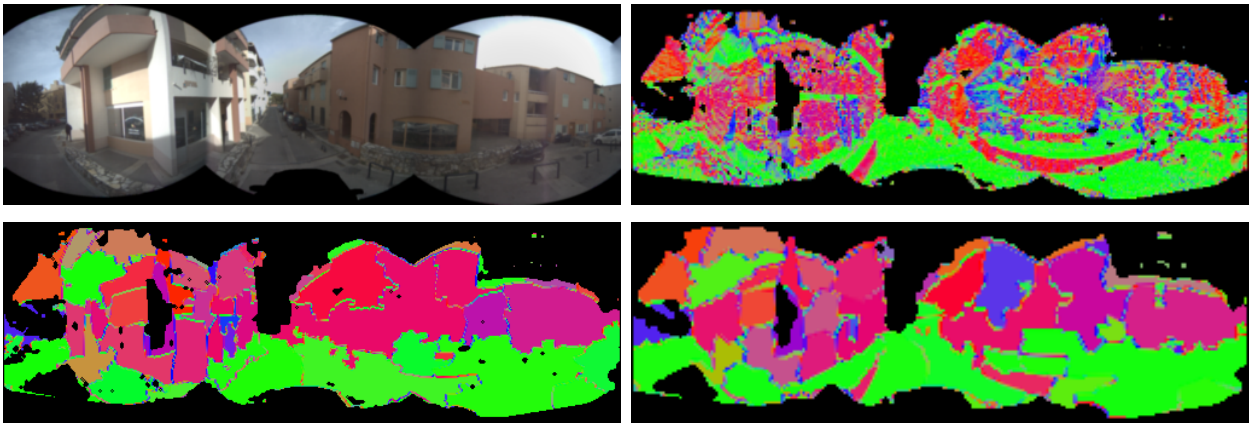

Figure 6.9 – Regularized depth using color and normal images. The color and low resolution normal images are show in the first row. The second row depicts the resulting regularization considering $\lambda = 0.1$ (left) and $\lambda = 1$ (right).

avoid over-segmentation in these areas. Some rendered views of the final regularized point clouds are given in figs. 6.8 and 6.9, for two outdoor frames with different levels of noise.

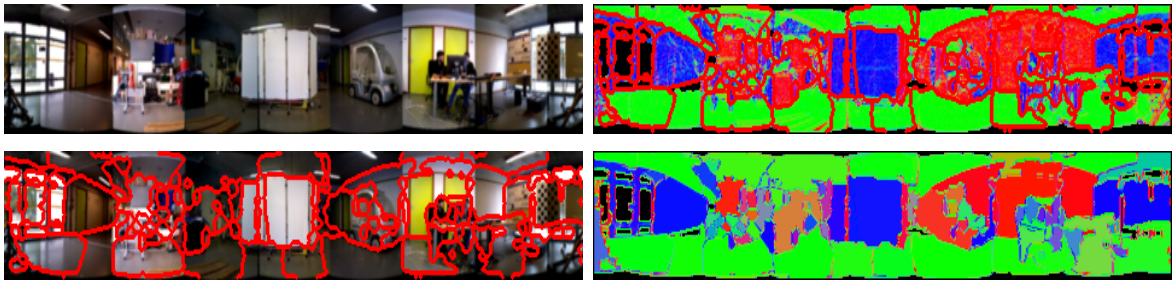

Figure 6.10 – OmniSLIC RGB-D image segmentation example for an indoor frame using $\lambda = 1$. The first row depicts the RGB and normal images. The second row depicts the resulting segmentation applied to the RGB image (left image) and the normals of the regularized depth (at right).

6.5.1 Outdoor Localization

In order to measure the effect of the regularization in the RGB-D registration, we performed the pose computation using an RGB-D sequence acquired in Garbejaire, as shown in fig. 6.11. The approximative real path of the camera is highlighted in blue and the starting-ending point is indicated by the green box. The trajectory in green considered the method of [Tykkala et al., 2011] with raw depth images. The same method but now considering regularized frames is shown in red and the adaptive RGB-D registration with regularized frames is shown in blue. All trajectories present drift in this long sequence, which is expected because no bundle adjustment or loop closure is performed. However, the accumulated trajectory errors are considerably smaller with the adaptive method and regularized frames, notably in the travelled distance. The approximate lengths of the real trajectory of the vehicle, without and with the regularization was respectively of 640, 302 and 583 meters. Besides the advantages of higher accuracy and robustness, we remark that there is also an advantage on the computational cost of the registration. Convergence is reached with a reduced number of iterations, and thus, the time required to register a pair of frames is around 40 % shorter in average with respect to the same optimization using the raw frames.

Finally, the improvement of the depth after the regularization is clearly apparent by visual inspection of the reconstructed point clouds in the Garbejaire sequence. Some examples are given in fig. 6.12. We can see on the left column the point cloud reconstructed from raw data and the same frame after regularization in the right. Notice how the artifacts and the waves on the floor are removed in the view after the proposed regularization. Inspection of the depth images and the normal vector images also confirm this, where flat and/or smooth surfaces such as the building façades or the road are more regular in the regularized images, whilst keeping semantically meaningful boundaries of the surfaces. We note that the scene can be roughly described with a small number of patches after the regularization. Although conceived for the stereo frames, the same procedure can be also applied to the indoor frames.

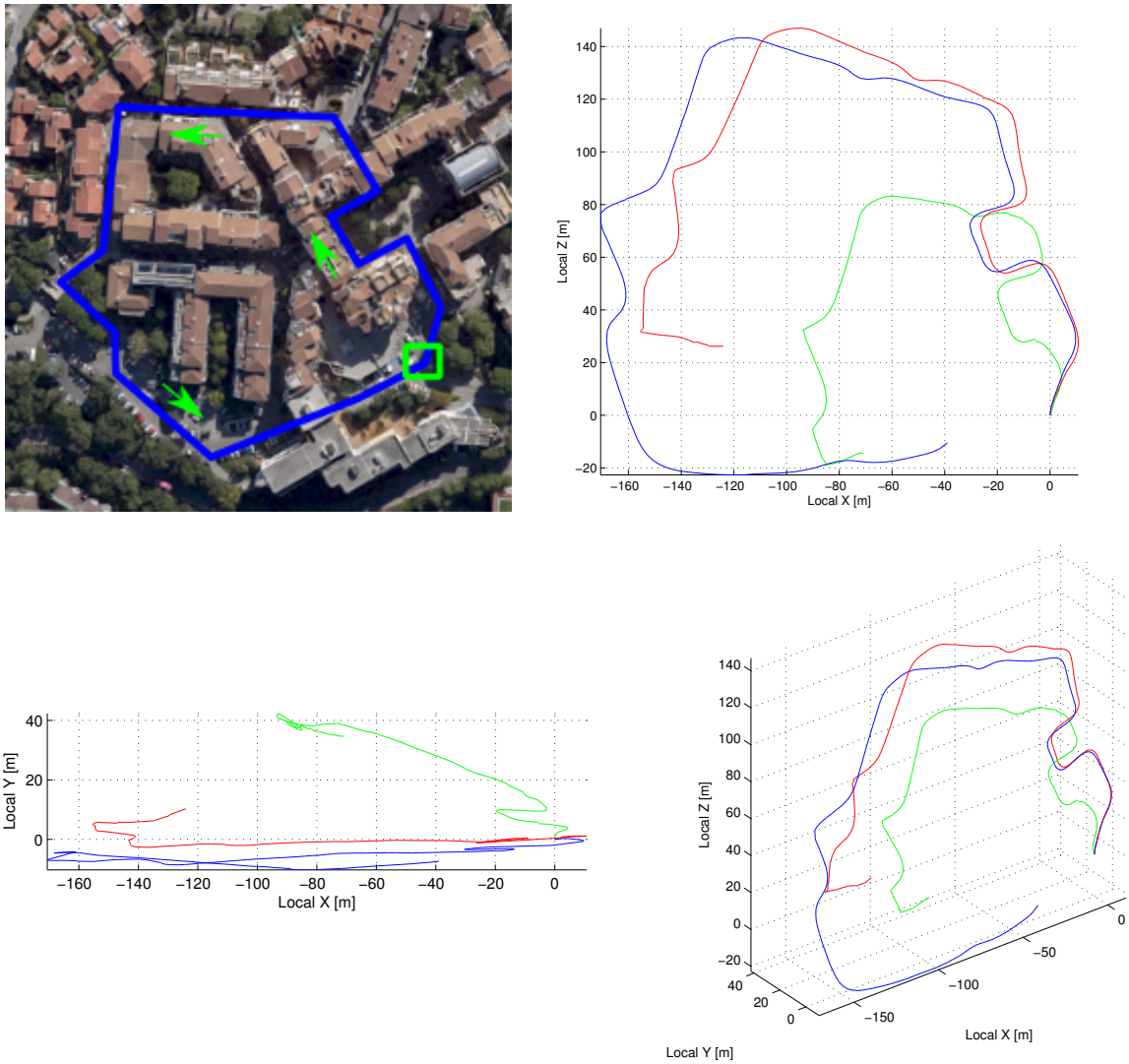

Figure 6.11 – Estimated trajectories for different direct tracking techniques using the Garbejaire sequence: [Tykkala et al., 2011] without regularization (in green), with regularization (in red) and the adaptive formulation of chapter 5 with regularization (in blue). The upper left figure is the top-view of the approximate trajectory and the starting-ending point is indicated by the green box. The remaining plots shows top and lateral views of the estimated trajectories. Note that the translation drift using the regularized frames is reduced.

6.6 Conclusions and Summary

We described in this chapter a method for the regularization of depth images using spatial and color constraints. This method is applied to our particular case of spherical vision, though it can be generally applied to other contexts evolving RGB-D data (as shown in the experiments section 6.5). We propose a modified version of the SLIC superpixel segmentation to wide FOV images and that uses the complementarity of color and surface orientation for the segmentation. This framework aims to correct the large errors induced by stereo matching specially in regular smooth surfaces while maintaining semantically meaningful boundaries. This formulation

increases the accuracy of the depth (e.g., by reducing the noise from wrong pixel block matching) and simultaneously enforces surface regularization. Furthermore, we extend the original RGB-D frames with an additional layer of information which encodes the uncertainty of each pixel. Subsequently, we applied this regularization framework to perform the localization of the camera in an outdoor environment. The resulting trajectories using the regularization have less drift, notably in the translation, indicating that the regularization improved the accuracy of direct registration and its robustness to errors coming from stereo matching.

Future research directions include exploring complementary constraints such as the orthogonality of the main planes (as in Manhattan World scenes) or the use of lines in the regularization. Another interesting point would be to explore the semantic information of the scene in the regularization.

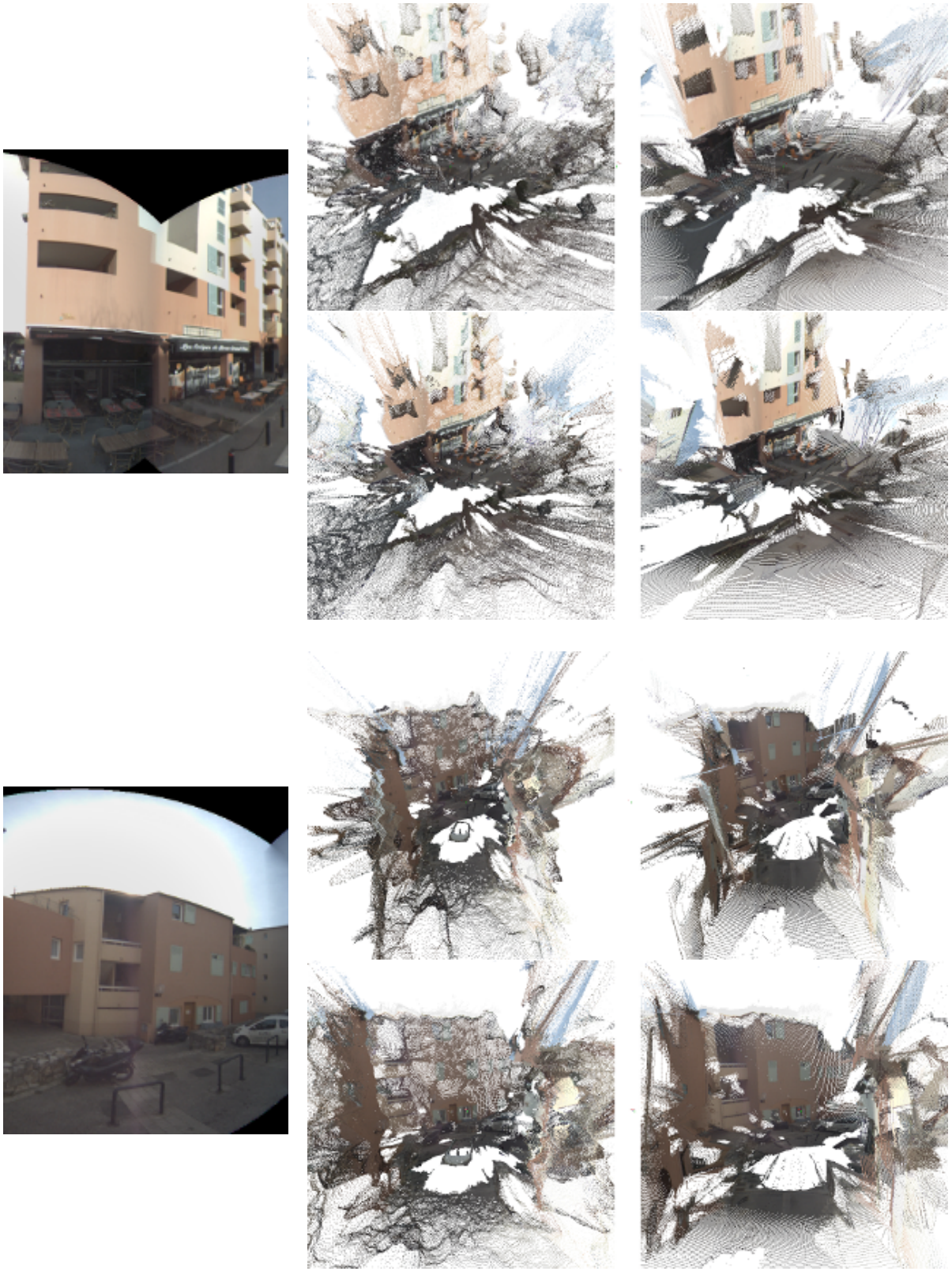

Figure 6.12 – Rendered point cloud views of two outdoor frames before (second column) and after the regularization (third column). The first column depicts the color image of the region of interest displayed in the rendered views. The last two rows are from the frame regularized displayed in fig. 6.9.

Chapter 7

RGB-D Compact Mapping for Optimal Environment Visibility

Contents

7.1 Introduction

Producing compact map representations of large-scale scenes is relevant for a wide number of applications, from autonomous navigation (e.g., [Whelan et al., 2015, Meilland et al., 2015]) to augmented reality and rendering (e.g., [Huhle et al., 2010, Anguelov et al., 2010]). For instance, building compact scene representations that can be accessed in constant time, regardless the environment size, is often a pre-requisite in mobile robotics applications such as robot localization and autonomous navigation (e.g., [Dayoub et al., 2011, Chapoulie et al., 2011, Meilland et al., 2015, Chiu et al., 2016]). In this context, RGB-D keyframe-based maps are a standard solution to produce compact representations from a continuous sequence of images. RGB-D keyframe-based techniques represent the world with a set of frames positioned in the scene, without performing an explicit reconstruction of the environment in a single coordinate system, as shown in fig. 7.1. This allows one to store a local photometric and geometric model of the scene for high accuracy tasks, while maintaining a topological framework at large-scale that is accurate enough to ensure the connectivity between the keyframes.

Ideally, the quality of a scene representation should be measured by the success of the end application. For instance, a good map representation for direct RGB-D registration can be unsuitable for monocular appearance-based tracking, feature-based methods or for image rendering. Therefore, the expected characteristics of each map are closely related to the specific user application. In this chapter, we describe a compact mapping scheme for localization and navigation tasks using direct RGB-D registration. Our goal is to find a compromise between the sparsity of the map while ensuring a good coverage of the environment for the registration techniques presented earlier in chapters 4 and 5. Furthermore, we want to create a unique map representation for different camera trajectories, which can be updated over time from the availability of new data of different explorations, as in the context of life-long mapping.

The chapter is organized as follows. Section 7.2 presents some related compact mapping approaches. Section 7.3 summarizes the stages for the keyframe selection. The camera tracking and free space segmentation are described in section 7.3.1. In section 7.3.2, we present possible strategies to select the locations of the keyframes to ensure a good coverage of the environment. In section 7.4, we present some preliminary compact mapping results using an indoor RGB-D sequence. Finally, we conclude the chapter in section 7.5.

7.2 Related Works

The robotics and computer vision communities have developed many techniques for 3D mapping from images and point clouds. Classic examples are the VSLAM frameworks with monocular cameras (e.g., [Silveira et al., 2008, Mur-Artal et al., 2015]), stereo cameras (e.g., [Anguelov et al., 2010, Whelan et al., 2015]), range/LIDAR and RGB-D cameras (e.g., [New-

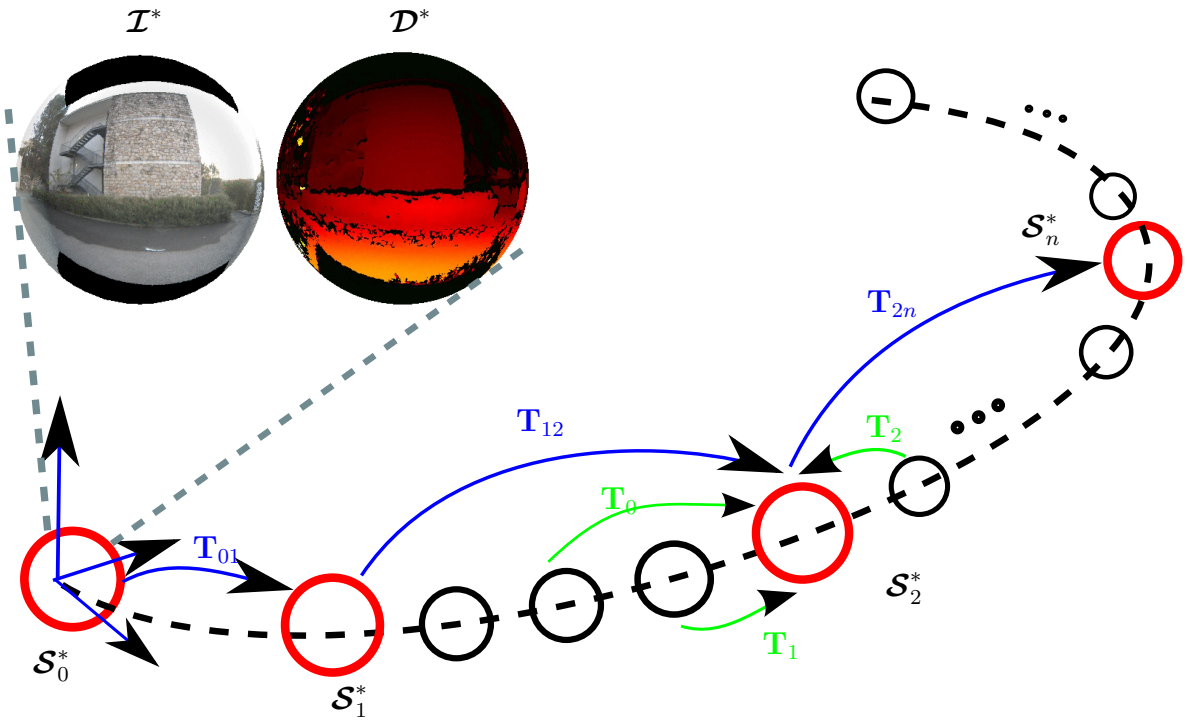

Figure 7.1 – Keyframe topo-metric map scheme: the nodes are RGB-D frames, which are linked by the relative poses $\mathbf{T} \in \mathbb{SE}(3)$. The poses $\mathbf{T}_{ij}$ link keyframes $\boldsymbol{\mathcal{S}}^*$ (in red), while $\mathbf{T}_k$ relate near frames (in black) to a particular keyframe. Only keyframes are kept in the final environment model. Each keyframe is built by exploiting the redundancy of nearby frames to reduce its noise and uncertainty.

combe et al., 2011a, Meilland and Comport, 2013, Dryanovski et al., 2013, Henry et al., 2014]) and even from unsorted collections of photos (e.g., [Snavely et al., 2006, Wu, 2013]). Most mapping systems require the spatial alignment of consecutive data frames, the detection of loop closures, and the pose refinement/fusion of individual frames. Recently, volumetric/voxel representations combined with truncated signed distance functions (TSDF) [Newcombe et al., 2011b, Newcombe et al., 2011a, Calakli and Taubin, 2011, Maier et al., 2015] have received widespread attention due to their reconstruction quality for creating accurate maps from multiple frames. For instance, to obtain geometrically accurate 3D reconstructions, [Newcombe et al., 2011a] fused RGB-D frames in a voxel-based representation and performed the tracking between individual frames and the fused/accumulated model with a point-to-plane ICP. In this chapter, however, we do not use this voxel-based representation since the storage capacity and computational burden are higher than performing the fusion of the depth and intensity directly on the RGB-D spherical keyframe.

Compact mapping deals with the problem of representing large-scale scenes with linear complexity. Classic techniques rely on efficient geometric discretizations to represent the scene without performing an explicit 3D reconstruction in a single global reference frame, as for

instance with topo-metric maps of perspective and spherical keyframes (e.g., [Dayoub et al., 2011, Chiu et al., 2016]) or segmenting the scene in piecewise planar patches (e.g., [Fernandez-Moral et al., 2013, Fernandez-Moral et al., 2016, Wang et al., 2016]). Compact mapping techniques are also pertinent to reduce the drift of visual odometry methods because performing frame to frame tracking introduces more drift in the trajectory due to the accumulation of optimization errors. Commonly used techniques to perform the keyframe selection are based on the down-sampling of the camera trajectory, combined with similarity distances as the median of absolute differences (MAD) of the intensity error (e.g., [Meilland et al., 2015]) or the pose covariance [Kerl et al., 2013b, Gokhool et al., 2015, Das and Waslander, 2015]. Of course other metrics can be considered to select these keyframes, such as, the number of iterations for convergence of the registration or the travelled distance between two frames (e.g., [Bachrach et al., 2012]). In this context, [Meilland et al., 2015] proposed a topo-metric representation composed of a graph of geolocalized RGB-D spherical images mainly designed for localization and autonomous navigation tasks. The RGB-D spherical frames are located in the scene using the direct registration method presented in chapter 5 with $\lambda = 0$. In order to limit drift inherent to visual odometry, only representative frames, along the trajectory, are kept in the map, based on the MAD of the photometric error. However, the images that were not selected as keyframes were dropped out and not exploited to improve the scene representation. This leads to losing useful information which could be used to enhance the quality of the RGB-D keyframes. In this chapter, we propose a framework which aims to integrate all the information available in a small number of RGB-D keyframes as depicted in fig. 7.1. Subsequently, our work presented in [Gokhool et al., 2015] proposed to filter the geometry of retained keyframes using a probabilistic average of nearby frames. The criteria to select new keyframes in the trajectory was based on the pose covariance to encode the entropy between two frames. This fusion formulation reduced the number of keyframes to represent the environment and also improved the quality of the map. However, the positioning of keyframes was closely related to the trajectory done by the camera. For example, a repetitive camera exploration inside an office would produce an unbounded number of keyframes using this strategy. Moreover, the fused keyframes maintained the same resolution and region-of-interest of the raw frames. In this chapter, we want to overpass these weaknesses by proposing a keyframe-based environment representation less sensitive to the trajectory undertaken by the camera. This is done by the introduction of a more elaborate keyframe positioning using the notions of visibility and accessibility of the environment. Furthermore, we want to filter and complete the information of the retained keyframes beyond the region-of-interest of the original frames.

7.3 Compact Keyframe Positioning Strategy

The idea to retain keyframes based on a predefined criteria proves to be very useful to produce a compact representation of the environment [Dayoub et al., 2011, Gokhool et al., 2015]. Hence, our compact mapping framework is based on a classical keyframe mapping strategy. The main goal is to build representative topo-metric models composed of a graph of geolocalized RGB-D keyframes optimally distributed in the scene. In the contexts of localization and autonomous navigation, the positioning of such keyframes can be done following two strategies. The first is of selecting representative locations along the trajectory of the camera during the exploration phase. In this case, the locations are chosen when a frame provide substantive new information from the previous keyframe, e.g., using either the MAD, pose covariance, traveled distance, number of iterations for convergence of the registration task, among others. The second strategy, which is described in detail in the following sections, is to explore jointly the appearance and topological information of the environment, such as visibility and accessibility. We present a partitioning of the environment free space using a Voronoi diagram. In summary, given an RGB-D sequence, the keyframe selection is decomposed in the following stages:

— The first stage is the computation of the relative pose between the frames following the temporal order. The computation is done from the direct camera tracking described in chapters 3 and 5.
— Subsequently, we perform the extraction of the local and accumulated free space. This stage is presented in section 7.3.1.
— The last stage is the selection of the keyframe positions using the free space partitioning (the vertices/bifurcation points of the Voronoi), which is described in section 7.3.2.

7.3.1 Free Space Extraction

Fortunately, a depth measurement provides more than just an observation on the surface location in the image. It also gives information about the free space between the surface and the camera. The "local free space" could be found as being the floor, which are the pixels with predominantly normals with $\mathbf{y}$ direction in the ground plane. The "total free space" is then the integration of the 3D points belonging to the "local free space" of the different frames. Afterwards, we project the 3D points belonging to the floor in a 2D grid and extract a free space probability distribution. The algorithm 7.3.1.1 describes in more detail the space extraction. Some examples of the local free space extraction using the low resolution images are shown in fig. 7.2.

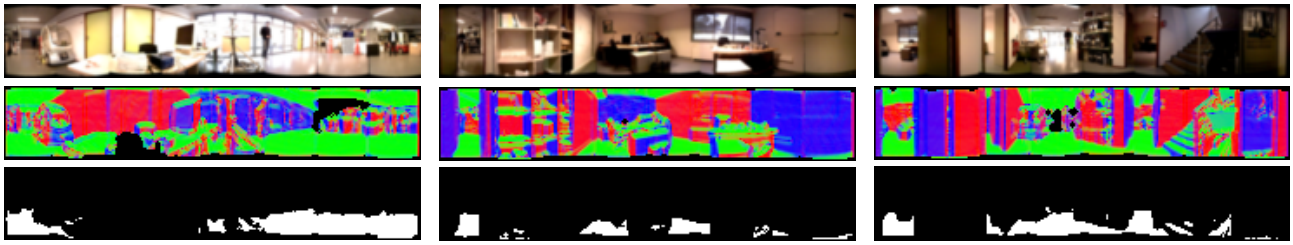

Figure 7.2 – Local free space extraction. Each column depicts the free space extraction of different frames. The top and middle rows show the RGB and surface normals respectively. The local free space is shown as the white pixels in the last row.

7.3.2 Space Partitioning and Optimal Coverage

Space partitioning/triangulation is a basic problem studied in computational geometry. Classic partitioning techniques use tessellations such as the Voronoi diagram [Ogniewicz and Ilg, 1992, Ogniewicz, 1994, Vázquez-Otero et al., 2015]. A good survey about describing shapes by using Voronoi diagrams is given in chapters 1 and 6 of [Siddiqi and Pizer, 2008]. This section

Algorithm 7.3.1.1 : Free Space Extraction

Inputs local free space : max normal angle and distance to the ground level: ε_θ and ε_d.

Inputs total free space : resolution and number of observations: r and n.
Output : boolean accumulated free space grid: **FG**.
List of 3D points in the free space: $\mathbf{LF} = [\,]$.
for all frames $\boldsymbol{\mathcal{S}}_i$ **do**
  Warp $\boldsymbol{\mathcal{S}}_i$ using its pose $\mathbf{T}_i$ and compute the surface normals: $\boldsymbol{\mathcal{S}}_{w_i}$ and $\mathbf{n}(\mathbf{p})$.
  for all pixels $\mathbf{p}$ in $\boldsymbol{\mathcal{S}}_{w_i}$ **do**
    Pixel not in the local free space: $\mathbf{F}(\mathbf{p}) = 0$
    Compute the angles of the normals with respect to the Y direction: $\Theta = \arccos((0\ 1\ 0)^T \mathbf{n}(\mathbf{p}))$.
    Compute the 3D point elevation: $d = (0\ 1\ 0)^T \boldsymbol{\mathcal{D}}_{w_i}(\mathbf{p})\Pi^{-1}(\mathbf{p})$.
    if $\|\Theta\|_1 < \varepsilon_\theta$ & $\|d\|_1 < \varepsilon_d$ **then**
      $\mathbf{LF} = [\mathbf{LF}\ \boldsymbol{\mathcal{D}}_{w_i}(\mathbf{p})\Pi^{-1}(\mathbf{p})]$ – point is inserted in the local free space.
    end if
  end for
end for
Find the convex hull of the free space $(x_{min}, x_{max}, z_{min}, z_{max})$: $x_{min} = \min(\mathbf{LF}(1,:)), x_{max} = \max(\mathbf{LF}(1,:)), z_{min} = \min(\mathbf{LF}(3,:))$ and $z_{max} = \max(\mathbf{LF}(3,:))$
Build 2D histogram edges: $E = \{(x_{min} : r : x_{max}), (z_{min} : r : z_{max})\}$.
$\mathbf{H} = hist3(\mathbf{LF}([1,3],:), 'Edges', E)$.
for all free space grid cells $\mathbf{i}$ in $\mathbf{H}$ **do**
  if $(\mathbf{H}(\mathbf{i}) > n)$ **then**
    $\mathbf{FG}(\mathbf{i}) = 1$.
  end if
end for

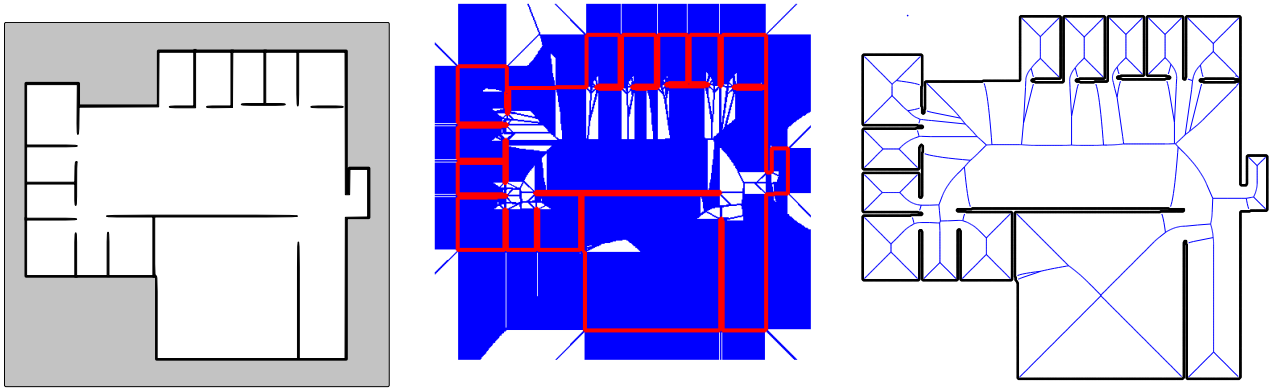

Figure 7.3 – CAD model of the ground floor of the Inria Kahn building and approximative free space region depicted in white (left image). The resulting Voronoi diagram with seed points (in red) are shown in the middle plot. The medial axes from the simplified Voronoi diagram are shown in blue (right image).

describes briefly Voronoi diagrams, followed by the partitioning framework of the extracted free space. One of the simplest application of Voronoi diagrams is the triangulation of a 2D space given n points (called seeds or generators) into a set of convex cells, where each cell contains exactly one generator point. This partitioning has an interesting property: for all points inside a cell, the distance to its seed point is smaller than to any other point. The edges of the cells are the equidistant straight lines (medial axis) between two seed points. The extremities of the edges are the vertices of the Voronoi (bifurcation points), which can be at infinity. The Voronoi can also be used to extract the topology of shapes, a process known as skeletonization [Siddiqi and Pizer, 2008, Vázquez-Otero et al., 2015]. The skeletonization process starts with finding the Voronoi from a discretization of shape contours (boundaries). This creates a complex diagram that can be simplified by pruning the edges whose vertices are out of the shape or with a threshold distance from the boundary. An example of this process in "simulation" using the floor CAD plan of the Inria building is given in fig. 7.3. As can be noticed, the resulting graph encapsulates the topology and accessibility of the different regions of the scene. Furthermore, the bifurcation points in this map reflects crossings regions.

How can this process be transposed to the partitioning of a real environment? The contours of the extracted free space, presented in the previous section, can be used to create this Voronoi diagram. We remark that each vertex (bifurcation of the diagram) defines a visibility radius in the scene, where the appearance model is "similar". Therefore, a plausible strategy is to position the keyframes in some of these vertices. Robots (and humans) often cross the environment in a neighborhood of the medial axes that define the vertices points. This is specially true in roads or corridors, where the trajectory of acquisition is often near the medial axis of the free space. In order to obtain a sparse model of representative keyframes, we select the most representative vertices of the diagram in terms of coverage and visibility. We start inserting a keyframe for the vertex point with the highest coverage and to sort the remaining vertices by their radius

of coverage. The subsequent vertices are then tested using an intersection operation. If their position lies inside the coverage of an already included vertex, they are not considered in the model, as presented in the algorithm 7.3.2.1.

7.3.3 Virtual Keyframe Rendering and Fusion

Once the locations of the keyframes are chosen, we can perform a fusion scheme such that each keyframe integrates the information from other nearby frames to improve its accuracy and completeness. This consists on warping the frames, the propagation of the uncertainty of the pose and of the depth and the fusion of the warped frames into the keyframe pose. The uncertainty propagation supposing Gaussian noise is described in the appendix B. Assuming that a set of n frames share information, their fusion corresponds to find $\boldsymbol{\mathcal{S}}$ that minimizes

Algorithm 7.3.2.1 : Space Partitioning and Optimal Coverage

Inputs : 2D free space grid and list of poses of the frames: **FG** and $LP = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_m\}$.

Output : List of keyframe locations: $T = \{v_1, v_2, v_3, ..., v_n\}$.
Compute the edges of the free space: $contours = edge(\mathbf{FG}, 'sobel')$.
Compute the Voronoi vertices: $[v_1, v_2, v_3, ..., v_k] = voronoi(contours)$.
List of valid vertices: $\mathbf{V} = [\,]$.
for all vertices v_k **do**
  Simplify the diagram from the free space and using distance to the border points:
  if v_k in free space **FG then**
    $\mathbf{V} = [\mathbf{V}\ v_k]$.
  end if
end for
Find the radius of visibility of each vertex:
for all vertices v_i in **V do**
  $radius(i) = \min(\|contours - v_k\|_2)$.
  Check if any frame was acquired inside this radius:
  list_of_frames(i) = find($\|LP - v_k\|_2 <= radius(i)$).
end for
List of keyframe positions: $T = [\,]$.
Sort vertices by their radius of coverage: sort(**V**).
for all vertices v_i in **V do**
  if T is empty **then**
    $T = v_i$.
  else if ($\|v_i - T\| < radius$ & list_of_frames(i) > 0) **then**
    $T = [T\ v_i]$.
  end if
end for

the weighted distance between the warped frames $\boldsymbol{\mathcal{S}}_{w_i} = \{\boldsymbol{\mathcal{I}}_{w_i}, \boldsymbol{\mathcal{D}}_{w_i}, \Sigma_{\mathcal{D}_{w_i}}\}$, $\boldsymbol{\mathcal{S}}(\mathbf{p}) = \text{argmin}_{\boldsymbol{\mathcal{S}}} \sum_{i=1}^{n} \Sigma^{-1}_{\mathcal{D}_{w_i}}(\mathbf{p})(\boldsymbol{\mathcal{S}}_{w_i}(\mathbf{p}) - \boldsymbol{\mathcal{S}}(\mathbf{p}))^2$, which can be stated sequentially as a weighted average as follows:

$$\begin{cases} \boldsymbol{\mathcal{I}}_{i+1}(\mathbf{p}) = \dfrac{\mathbf{W}_i^D(\mathbf{p})\boldsymbol{\mathcal{I}}_i(\mathbf{p}) + \Sigma^{-1}_{\mathcal{D}_{w_i}}(\mathbf{p})\boldsymbol{\mathcal{I}}_{w_i}(\mathbf{p})}{\mathbf{W}_i^D(\mathbf{p}) + \Sigma^{-1}_{\mathcal{D}_{w_i}}(\mathbf{p})} \\ \boldsymbol{\mathcal{D}}_{i+1}(\mathbf{p}) = \dfrac{\mathbf{W}_i^D(\mathbf{p})\boldsymbol{\mathcal{D}}_i(\mathbf{p}) + \Sigma^{-1}_{\mathcal{D}_{w_i}}(\mathbf{p})\boldsymbol{\mathcal{D}}_{w_i}(\mathbf{p})}{\mathbf{W}_i^D(\mathbf{p}) + \Sigma^{-1}_{\mathcal{D}_{w_i}}(\mathbf{p})} \\ \mathbf{W}_{i+1}^D = \mathbf{W}_i^D + \Sigma^{-1}_{\mathcal{D}_{w_i}} \end{cases} \tag{7.1}$$

where, $\Sigma^{-1}_{\mathcal{D}_{w_i}}$ is the confidence (the inverse of the uncertainty) resulting from the blending of both pose and structure errors as described in section B.1; and $\mathbf{W}_0^D$, being the uncertainty of the initial keyframe model. In the presence of a large number of frames, one can consider a more robust fusion to outliers and occlusions, such as the median [Merrell et al., 2007]:

$$\begin{cases} \boldsymbol{\mathcal{I}}(\mathbf{p}) = \text{median}(\boldsymbol{\mathcal{I}}_{w_1}(\mathbf{p}), \boldsymbol{\mathcal{I}}_{w_2}(\mathbf{p}), ..., \boldsymbol{\mathcal{I}}_{w_n}(\mathbf{p})) \\ \boldsymbol{\mathcal{D}}(\mathbf{p}) = \text{median}(\boldsymbol{\mathcal{D}}_{w_1}(\mathbf{p}), \boldsymbol{\mathcal{D}}_{w_2}(\mathbf{p}), ..., \boldsymbol{\mathcal{D}}_{w_n}(\mathbf{p})). \end{cases} \tag{7.2}$$

7.4 Experiments

In this section, we present preliminary results for the compact mapping of the indoor sequence Inria2, shown in chapter 5. We evaluate the mapping quality in terms of its compactness (the number of retained keyframes) and of the accuracy of direct registration using the virtual keyframes. The first stage of the mapping pipeline is the localization of the camera using the images. The initialization and the adaptive registration techniques were used to perform this localization. For further increasing the pose estimation accuracy and to reduce the drift in the trajectory, we performed the optimization of the poses using a loop closure between the first and last frames. The initialization technique was used to compute a rough pose between these frames, which were subsequently refined by the direct method. The loop closure pose is used as edge for the trajectory refinement with the GTSAM factor graphs optimization library [Dellaert, 2012], which results in the optimized trajectory shown in fig. 7.4.

Once the trajectory is estimated, we extracted the local free space of each frame, as shown in fig. 7.4. The accumulated free space is composed of the "local free space", represented by white pixels in the image. We subsample the points belonging to the floor using a 2D grid with $r = 0.1$ meters of resolution. Occluded or non-visited areas are depicted in gray. The borders

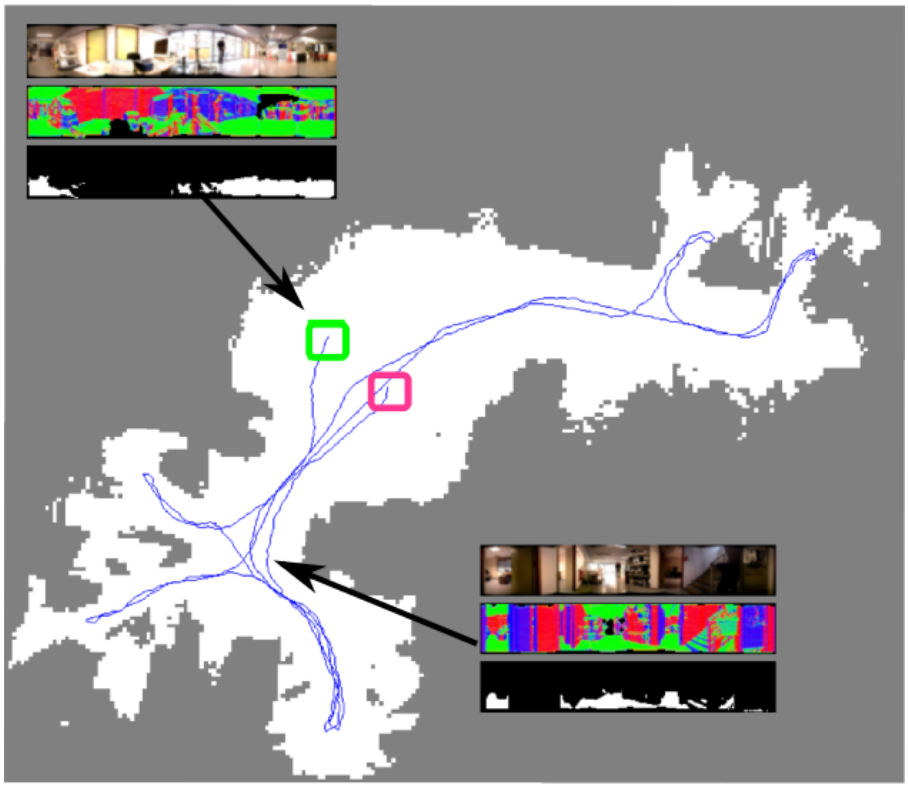

Figure 7.4 – Free space extraction in the real indoor sequence. The camera trajectory starts in the green box and the endpoint is indicated by the red box. The points belonging to the floor in each frame are used to extract the free space. Two examples of the regions of the free space are shown with the RGB and normal images. The accumulated free space grid using the computed trajectory are the regions in white, while the occupied or not explored regions are shown in gray.

of the free space are used to build the Voronoi diagram using the algorithms 7.3.1.1 and 7.3.2.1, as shown in fig. 7.5. The medial axes of the scene and its vertices using the diagram are shown in the right plot.

This strategy allowed a convenient sparse representation of the environment as depicted in fig. 7.6, where only 27 nodes are retained. Interestingly, the coverage of the free space is of approximately 89% with these 27 vertices, which represents less than 1% of the number of available frames (27/3250). For a comparison of keyframe sparsity, using the entropy and the probabilistic fusion [Gokhool et al., 2015], the number of keyframes was 67 for the first pass of the trajectory. We can clearly identify some regions in the scene with higher density of keyframes as the areas where the geometric appearance of scene changes, such as in crossing regions with partial occlusion. The sparsity of the keyframes is notably observed in the regions of the scene with invariant viewing conditions (i.e., with a predominantly convex geometry) and the distance of retained frames was of up to 3.1 meters, which is approximate three times bigger than without this partitioning technique. We reinforce that the sparsity of the map might be adapted to the capacities of the posterior registration algorithm to explore it. The

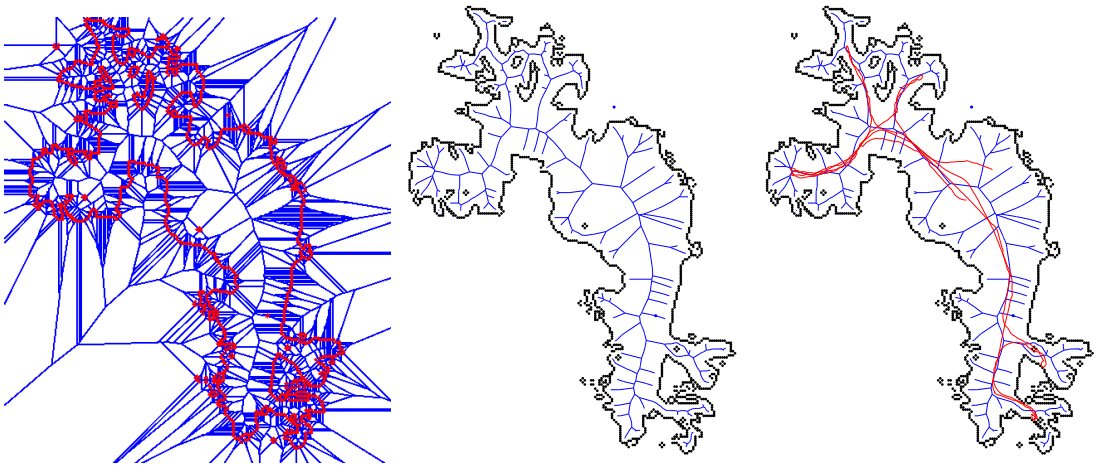

Figure 7.5 – Voronoi and optimal scene coverage in the real indoor sequence. The raw Voronoi diagram is shown in the left image, the resulting segmentation of the scene is obtained after simplifying the diagram (blue lines in the middle plot). The simplified Voronoi edges and the trajectory of the camera (in red) are shown superposed in the right image. In this indoor sequence, the robot was driven near the principal medial axes of the scene.

keyframes are also more accurate and complete, as shown in fig. 7.7. Two examples of virtual keyframes with a smaller visibility coverage can be seen in fig. 7.8.

7.4.1 Direct Registration Using Virtual Keyframes

We evaluate the coverage of the environment, the accuracy and completeness of the scene model by performing registration experiments using some virtually rendered keyframes. In this experiment, we registered a different sequence of images lying inside the coverage radius of the keyframe. For a preliminary baseline comparison, we performed the keyframe selection and rendering using the following strategies: i) using as keyframe the nearest raw depth image to the vertex, as in [Meilland et al., 2015]; ii) using the presented formulation and placing the keyframe in the vertex, combining nearby frames and increasing the region-of-interest of the depth and intensity images. Table 7.1 shows the average errors obtained by the different keyframe models using the initialization and adaptive registration techniques. A benefit of the improved keyframes is the higher accuracy of direct registration as a consequence of the higher accuracy of the virtual keyframe. The region of convergence also enlarges as can be noticed in the convergence rate in the last column of Table 7.1. We assumed that the algorithm converged in cases where the estimation error is smaller than 7 degrees in the rotation and of 10 centimeters in the translation.

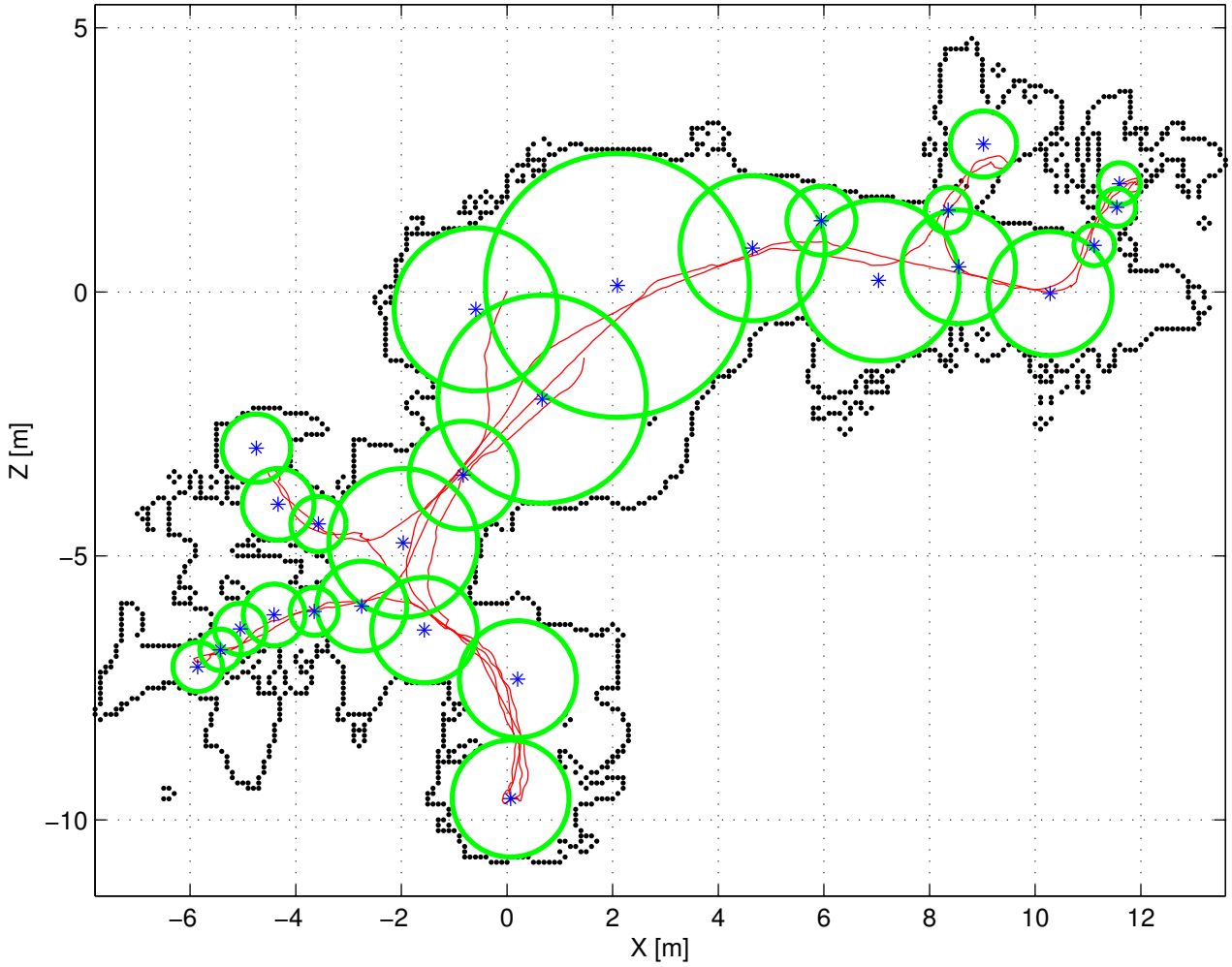

Figure 7.6 – Vertices pruning with the criteria of visibility and coverage. The environment can be sparsely represented with frames in 27 positions (blue $*$). The radius of visibility coverage for each vertex is indicated by the green circles.

Figure 7.7 – Virtual keyframe fusion example. The left image depicts the initial keyframe. The resulting keyframe after fusion of the nearby frames is shown in the right. Observe that the occlusions are handled using this representation and the final model is a more complete keyframe with a bigger region of interest.

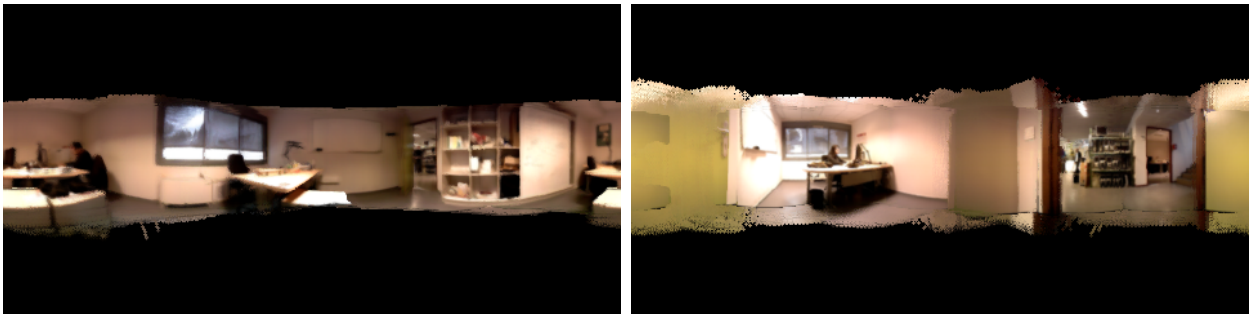

Figure 7.8 – Virtual keyframe examples with smaller coverage radius. The first virtual keyframe from the left depicts the office with few nearby frames included in the radius of visibility. The second image is the resulting keyframe in a door crossing, which also has a small radius of visibility.

Table 7.1 – Convergence and average registration errors using different keyframe models.

	Av. Rot. Error (deg)	Av. Trans. Error (mm)	Convergence
Raw Frame	12	170	57%
Virtual Keyframe	2.2	30	92%

7.4.2 Discussion

Ideally, a compact map guaranteeing visibility would consist of full spherical frames positioned in all bifurcation points. This would ensure a coverage of the scene with maximal visibility. The free space partitioning around the principal medial axes is a strategy to reduce the number of virtual keyframes, while maintaining sufficient visibility conditions in the model. For instance, in the map of the indoor sequence presented in section 7.4, only 27 frames were kept resulting in a coverage of the environment of around 90%. A number of perspectives can be drawn from these preliminary results. First, a unique and stable map representation can be defined using the visibility constraints. This unique representation allows the update of the keyframes in the map over time, as new information about the scene is acquired. The concepts of long term and short term memories could be explored in this sense to update the photometric and geometric model of the keyframes. Second, this representation maximizes the visibility of retained keyframes in the scene. Notably, keyframes are located in crossing regions which are critical for appearance-based registration techniques. However, some issues might appear while rendering the keyframes in a distant viewpoint. Particularly, the level of detail of the rendered RGB images is reduced due to the noise of depth, pose and the calibration errors in the image stitching. To reduce this effect, super-resolution could be used to increase the sharpness of the RGB images in the virtual keyframes. We also note that different metrics could be used to extract the free space, to select the keyframe locations and the nearby frames. Notably, a subsampling of the Voronoi edges, as the middle points between two vertices or the intersection of the coverage circles with the medial axes. We leave these considerations as future research.

7.5 Conclusions

This chapter presented a compact mapping technique based on keyframes. We discussed the positioning of the keyframes in the scene from two points of view: down-sampling the trajectory of acquisition and down-sampling the free space. Note that our objective is to build sparse keyframe-based maps that will be used for posterior localization and navigation tasks. Therefore, the environment model must take into account the capacities of the posterior registration techniques, specially in terms of basin of convergence and visibility. One critical aspect while using our initialization and adaptive direct methods is the visibility of the frames, since registering frames with minimal visibility often leads to divergence of the registration. An interesting idea for maintaining visibility in a sparse keyframe representation is the partitioning of the free space to emphasize appearance and topological properties of the environment. Our preliminary results suggest that a sparse keyframe map that has good visibility properties is, in this sense, a convenient compact model. Furthermore, this model can be used with different camera trajectories and be updated over time from the availability of new data, as in the context of life-long mapping.

Chapter 8

Conclusions and Perspectives

This thesis addressed the problem of pose estimation and mapping from RGB-D images. While most existing registration methods are based on the extraction and matching of characteristics (feature-based), this thesis focused on parametric direct (appearance-based) techniques to perform the camera tracking and to build a dense keyframe-based map model of the scene. Direct methods are known for their subpixel accuracy and robustness to outliers in image alignment, but with the restriction of having small camera motions or high frame-rate. The main contributions of this work are the strategies to relax this restriction of small motions between RGB-D images. Notably, we present strategies that explored conveniently the photometric and geometric information, while maintaining the accuracy of direct image registration. In this context, we proposed a fast pose estimation technique to compute a rough estimate of large motions between depth images, which is used with an initialization scheme to image registration. This pose estimation is divided in two decoupled stages, the rotation and then the translation estimation, both based on the normal vectors orientation and on the depth. These two stages are efficiently computed from distributions using low resolution depth images. The limitations and observability of this pose computation have also been analyzed. Subsequently, we proposed an adaptive registration approach to improve the basin of convergence by shaping the cost function. This approach explored the observations that the intensity and depth error terms display different convergence properties for small and large motions. Experiments using spherical and perspective images indicate that these methods present substantial improvements in the convergence of the camera tracking, enabling to efficiently align images rotated of 180 degrees and with translations up to three meters for the spherical images.

In the second part of the thesis, we have treated the problem of building a useful and sparse representation of the scene. Producing this model is performed offline and therefore, we can apply different prepossessing to the image frames. In this context, we presented a regularization approach to filter the stereo depth images. Notably, we proposed an extension of the state-of-the-art SLIC superpixel algorithm to segment the frames leveraging geometric information such as surface orientation and color. We also proposed an adapted version of

SLIC to omnidirectional images by using the geodesic in the sphere, instead of the Euclidean norm. Finally, we described a compact mapping framework to create sparse topo-metric RGB-D maps of the scene. The map is composed of keyframes optimally distributed along the environment, using the notions of visibility and space coverage. The developed registration and regularization approaches are exploited to build this sparse keyframe map model of the environment. The locations of the keyframes use a convenient partitioning of the free space of the scene. Our preliminary results suggest that this sparse keyframe map has good visibility properties and therefore is a convenient compact model to be used with appearance-based registration techniques. Notably, using this space partitioning, less than 1% of the frames have a coverage of more than 90% of the environment. Furthermore, this map can be used with different camera trajectories and be updated over time from the availability of new data, as in the context of life-long mapping. We show the effectiveness of these approaches in localization and mapping experiments of indoor and outdoor real scenes, where the compactness, accuracy and consistency of the maps were greatly improved.

The techniques presented in this thesis allowed to handle larger separation between frames, and thus, lower camera frame rates and/or higher camera speed can be attained. This has also a direct impact on the hardware resources, reducing the computation and memory requirements as less frames are processed for localization and mapping purposes.

8.1 Future Work

We have a diverse set of research directions to be further explored. One interesting line of research is to design the hybrid RGB-D cost function for direct registration with theoretical guarantees of convergence. The adaptive RGB-D registration presented in chapter 5 improved the basin of convergence by adapting the photometric and geometric scale factor during the optimization. However, the design of the scaling strategies were supported mainly by empirical observation and by the shapes of level curves of the cost functions. It would be pertinent to further analyze theoretical properties of these costs. Notably, this line of research is correlated to the problem of defining the convergence domain of the different cost functions, i.e., the definition of upper bounds of convergence for direct image registration, in analogy to the "learned model" established in [Churchill et al., 2015, Dequaire et al., 2016] for feature-based registration techniques.

The good performance of the presented formulations was in part possible thanks to the large FOV provided by the spherical images and its resulting properties. With small FOV sensors and large motions, the frames might not have co-visibility and we cannot estimate the pose. Still, some important properties of spherical images were not explicitly exploited for pose estimation. For instance, the relation between a point and its antipodal in the sphere [Lim and Barnes, 2008, Corke and Mahony, 2009] could be used in the adaptive formulation in chapter

5. The pose estimation from normals in chapter 4 did not use motion properties in the sphere to eventually select the right mode in the distribution, or even build the distribution in a more convenient way. Another possibility to increase the robustness of this method would be using simultaneously intensity and color information to find the overlapped regions.

In terms of the frame regularization presented in chapter 6, we have considered a simple regularization technique which acts mainly as a low-pass filter, constrained by an intermediary superpixel segmentation of geometric and photometric surface edges. However, other appearance terms could be added in the regularization. Examples include constraints from prior knowledge of the environment (e.g. from orthogonality, image lines, main directions) or using higher level information such as the semantic segmentation of the different surfaces.

In the compact map front, different metrics could be used to extract the free space, to select the keyframe locations and the nearby frames, described in chapter 7. For instance, an additional subsampling of the Voronoi edges, as the middle points between two vertices or the intersection of the coverage circles with the medial axes could increase the visibility conditions. We also note that the concepts of long term and short term memories could be explored to update the photometric and geometric model of the keyframes. Another interesting topic is of using active perception while performing the acquisition of the sequence of RGB-D images. The trajectory of the mobile platform used to acquire the RGB-D sequences did not consider the observability conditions of the scene. For instance, the pose uncertainty of following a path with highly textured surfaces is reduced when compared to following a path passing through a single plane without texture with direct registration. In this context, online perception-aware path planning (e.g., [Salaris et al., 2017, Costante et al., 2016]) adapted to image registration would select the more appropriate path to reduce the trajectory uncertainty and therefore improving the scene photometric and geometric models.

Finally, we remark the rapid development of formulations employing convolutional neural networks (CNN) for scene segmentation, but also for motion estimation, tracking and mapping with promising initial results. For instance, we note [Kendall et al., 2015] and [Weyand et al., 2016] for localization from images, [Konda and Memisevic, 2015, Nicolai et al., 2016, Melekhov et al., 2017] for relative pose estimation from depth/RGB-D images and [Dosovitskiy et al., 2015, Guney and Geiger, 2016] for optical flow and disparity computation. These approaches employ end-to-end CNNs for both pose and flow/depth estimation, i.e., the neural network handles simultaneously the correspondence and the pose or flow computation. Exploring the capabilities of deep learning techniques as a tool for identifying relevant information, computing the camera motion and scene structure might be an interesting topic, specially in the context of large camera motions treated in this thesis.

Conclusions et Perspectives

Cette thèse a proposé de nouvelles contributions au problème de l'estimation de pose et de cartographie à partir d'images RGB-D. Bien que la plupart des méthodes d'enregistrement existantes soient basées sur l'extraction et la mise en correspondance d'éléments caractéristiques (features) de l'image, cette thèse s'est concentrée sur des techniques paramétriques directes (basées sur l'apparence) pour estimer la pose de la caméra et pour construire une représentation dense d'environnements à grande échelle. Les approches directes sont connues pour leur précision sous-pixellique et leur robustesse aux outliers dans les méthodes d'alignement d'images, mais sous la contrainte de petits déplacements de la caméra ou d'une fréquence d'acquisition d'image élevée. Les principales contributions de ce travail sont les approches méthodologiques permettant de relâcher cette restriction aux petits déplacements dans le cas d'images RGB-D. Notamment, nous présentons des méthodes permettant d'exploiter les différences de convergence des erreurs photométrique et géométrique, tout en maintenant la précision de l'enregistrement direct de l'image. Dans ce contexte, nous avons proposé une technique d'estimation de pose rapide pour calculer une estimation approximative des mouvements importants entre les images en profondeur, qui peut être utilisée comme initialisation des méthodes directes. Cette estimation de pose s'appuie sur les propriétés d'invariance en rotation des images sphériques, pour traiter de façon séquentielle l'estimation de la rotation puis de la translation en utilisant à la fois l'orientation des vecteurs normaux aux plans de la scène et l'information de profondeur. Ces deux étapes sont calculées efficacement à partir de distributions utilisant des images en profondeur à basse résolution. Les limites de validité de la méthode d'estimation de pose ont également été analysées notamment en terme d'observabilité. Par la suite, nous avons proposé une approche d'enregistrement d'images permettant d'améliorer le bassin de convergence en modifiant la fonction de coût de façon adaptative. Cette approche exploite les propriétés de convergence différentes des termes d'erreur d'intensité et de profondeur en fonction de l'amplitude des mouvements de la caméra. Les résultats expérimentaux utilisant des images sphériques et perspectives confirment que ces méthodes apportent des améliorations substantielles dans la convergence du suivi de la caméra en permettant d'aligner efficacement les images avec des rotations de 180 degrés et des translations jusqu'à trois mètres dans le cas des images sphériques.

Dans la deuxième partie de la thèse, nous avons traité le problème de la construction d'une représentation compacte et garantissant une bonne couverture de la scène. La construction de

ce modèle étant effectuée hors ligne, différents pré-traitements sont appliqués aux images. Une méthode de régularisation est proposée pour filtrer les images de profondeur issues du capteur sphérique stéréo. Cette méthode repose sur une extension de l'algorithme de superpixel SLIC pour segmenter les images RGB-D en exploitant les informations d'orientation de la surface et de couleur. Nous avons également proposé une version adaptée de SLIC à des images panoramiques en utilisant la géodésique dans la sphère, au lieu de la norme euclidienne. Enfin, nous avons décrit un cadre de cartographie compact pour créer des cartes RGB-D topo-métriques de la scène. La carte est composée d'images clés RGB-D réparties de façon optimale dans l'environnement, en utilisant les notions de visibilité et de couverture spatiale. Les approches développées d'enregistrement et de régularisation sont exploitées pour construire cette carte. Le positionnement des images clés dans la scène est réalisé à partir du Voronoï calculé sur l'espace libre. Nos résultats préliminaires montrent que cette carte topo-métrique possède de bonnes propriétés de visibilité et de compacité et, de ce fait, est particulièrement bien adaptée aux approches denses d'enregistrement basées sur l'apparence. Notamment, en utilisant cette représentation, il est possible de couvrir 90% de l'environnement avec seulement 1% des images images sphériques RGB-D construites à partir des données d'acquisition. En outre, cette carte peut être mise à jour au fil du temps en intégrant de nouvelles données acquises avec différentes trajectoires de caméras. Cette capacité présente un intérêt majeur dans le contexte de la cartographie à long-terme et de la navigation autonome. L'intérêt de ces approches est montré par les résultats d'expérimentation de localisation et de cartographie sur des scènes réelles d'intérieur et d'extérieur, où la compacité, la précision et la cohérence des cartes ont été considérablement améliorées.

Les méthodes développées dans cette thèse visent à étendre le champ d'application des approches d'enregistrement basées apparence et à proposer des représentations compactes de l'environnement bien adaptée aux problématiques de robotique mobile et de cartographie dense. Ces méthodes permettent de gérer des séquences d'acquisition où les déplacements entre les images sont importants comme , par exemple, dans le cas de séquences acquises par des caméras embarquées sur des véhicules à forte dynamique (drones, voitures autonomes). Cela a également un impact direct sur les ressources computationelles, notamment dans le cas d'applications temps réel embarquées, en réduisant les besoins en calcul et en mémoire par le fait que moins de trames sont traitées pour la localisation et la cartographie. Enfin, les méthodes proposées amènent directement à des représentations efficaces de la scène adaptées à la cartographie d'environnement à grande échelle et apte à intégrer de nouvelles données d'acquisition (Life long learning).

1 Travaux Futurs

Ces travaux de thèse ouvrent sur de nouvelles voies de recherche à approfondir. Parmi celles-ci, il parait important de pouvoir établir une garantie théorique de convergence pour les méthodes d'enregistrement direct utilisant des fonctions de coût hybride photométrique et géométrique. L'enregistrement RGB-D adaptatif présenté dans le chapitre 5 a amélioré le bassin de convergence en adaptant le facteur d'échelle photométrique et géométrique lors de l'optimisation. Cependant, le développement de notre approche vient principalement de l'observation empirique des courbes de niveau des fonctions de coût. Il serait pertinent d'analyser davantage les propriétés théoriques de ces fonctions de coûts. Une piste de recherche serait d'étendre la notion de "modèle appris" proposé dans [Churchill et al., 2015, Dequaire et al., 2016] pour les techniques d'enregistrement basées sur les features, pour établir une définition des limites supérieures de convergence dans le cas des approche basées apparence.

Les bonnes performances des méthodes présentées reposent en partie sur le large champ de vue fourni par les images sphériques et les propriétés qui en résultent. Avec des capteurs avec FOV réduits et de grands mouvements de la caméra, les images peuvent ne pas avoir une co-visibilité et, dans ce cas, ne pas permettre d'estimer la pose. Pourtant, certaines propriétés importantes des images sphériques n'ont pas été explicitement exploitées pour l'estimation de pose. Par exemple, la relation entre un point et son antipodal dans la sphère [Lim and Barnes, 2008, Corke and Mahony, 2009] pourrait être utilisée dans la formulation adaptative présentée dans le chapitre 5. L'estimation de la pose à partir des normales dans le chapitre 4 n'a pas utilisé les la possibilité de détecter les objets dynamiques dans la sphère pour robustifier la sélection du bon mode dans la distribution. Une autre possibilité d'augmenter la robustesse de cette méthode serait d'utiliser simultanément l'intensité et les informations de couleur pour trouver les régions superposées.

En ce qui concerne la régularisation des sphères RGB-D présentée dans le chapitre 6, nous avons considéré une technique de régularisation simple qui agit principalement comme un filtre passe-bas, contraint par une segmentation intermédiaire en superpixel des bords de surface géométriques et photométriques. Cependant, d'autres termes d'apparence pourraient être ajoutés dans la régularisation issus des contraintes liées à la connaissance préalable de l'environnement (par exemple, de l'orthogonalité, des lignes d'image, des directions principales) ou de l'utilisation d'informations de haut niveau telles que la segmentation sémantique des différentes surfaces.

Concernant la création de cartes 3D compactes décrite dans le chapitre 7, d'autres mesures pourraient être utilisées pour extraire l'espace libre, pour sélectionner les emplacements des images clés et les images proches. Rajouter des images clés à des endroits particuliers sur les branches du Voronoï, comme les points du milieu entre deux noeuds ou l'intersection des cercles de couverture avec les axes médians augmenterait les conditions de visibilité. Les concepts de

mémoires à long terme et à court terme demandent également à être explorés pour mettre à jour le modèle photométrique et géométrique des images clés. Dans nos approches, la trajectoire de la plate-forme mobile utilisée pour acquérir les séquences RGB-D ne prend pas en compte les conditions d'observation de la scène. En contrôlant le mouvement de la caméra par des approches de perception active lors de l'acquisition de la séquence d'images RGB-D, il serait possible d'améliorer les résultats. Par exemple, l'incertitude créée dans la pose en suivant un chemin avec des surfaces fortement texturées est réduite par rapport à suivre un chemin passant par un seul plan sans texture qui contraint peu les méthodes d'enregistrement direct. Dans ce contexte, la planification de chemin en ligne (par exemple, [Salaris et al., 2017, Costante et al., 2016]) permettrait de sélectionner la trajectoire la plus appropriée pour réduire l'incertitude de pose et donc améliorer les modèles photométriques et géométriques de la scène.

Enfin, le développement rapide, et avec des résultats prometteurs, des formulations utilisant des réseaux des neurones convolutionnels (CNN) pour la segmentation des scènes, mais aussi pour l'estimation du mouvement, le suivi et la cartographie, ouvre de nouvelles voies de recherche. Par exemple, [Kendall et al., 2015] et [Weyand et al., 2016] pour la localisation à partir d'images, [Konda and Memisevic, 2015, Nicolai et al., 2016, Melekhov et al., 2017] pour une estimation de pose relative à partir d'images en profondeur/RGB-D et [Dosovitskiy et al., 2015, Guney and Geiger, 2016] pour le flux optique et le calcul des disparités, apportent de nouvelles formulations à ces problèmes. Le réseau neuronal gère simultanément la correspondance et le calcul de pose ou de flux optique. L'exploration des capacités des techniques de "deep learning" comme un outil pour identifier les informations pertinentes, le calcul du mouvement de la caméra et de la structure des scènes sont des approches prometteuses, en particulier dans le contexte des caméras soumises à de fort déplacements qui est au centre de ce travail de thèse.

Appendices

Appendix A

Photometric and Geometric Jacobians

A.1 Photometric Error Jacobians

We adopt the convention of the right side group multiplication for computing the Jacobians, we refer to [Blanco, 2010] (sections 10.2 and 10.3) for the left side multiplication parametrization. The photometric error between the current and reference frames is:

$$e_I(\mathbf{p}, \mathbf{x}) = \boldsymbol{\mathcal{I}}(w(\mathbf{p}, \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}). \tag{A.1}$$

Due to due to group properties, the photometric Jacobian $\mathbf{J}^I$ (1x6) can be decomposed as:

$$\mathbf{J}^I = \nabla_{\mathbf{p}}(\boldsymbol{\mathcal{I}}(w(\mathbf{p}, \hat{\mathbf{T}})))\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{T}}, \tag{A.2}$$

where $\nabla_{\mathbf{p}}$ (1x3) is the image gradient with zero in the 3rd component. $\mathbf{J}_{\mathbf{T}}$ (12x6) is the Jacobian of the rigid transformation $\mathbf{T}(\mathbf{x})$ relative to the instantaneous velocities $\mathbf{x}$. Each Jacobian row is composed of the flatten versions of 3 first rows of the generators $\mathbf{A}_i$ shown in (3.18) of section 3.3.3.1:

$$\mathbf{J}_{\mathbf{T}} = \begin{pmatrix} \text{flatten}(\mathbf{A}_1) \\ \dots \\ \text{flatten}(\mathbf{A}_6) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \dots & & & & & \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{A.3}$$

Afterwards, $\mathbf{J}_{\mathbf{w}}$ (3x12) is the Jacobian of the warping function which depends on the sensor projection model. Given the 3D point $\mathbf{P}(\mathbf{p}) = (x\ y\ z)^T$ and considering squared pixels, the Jacobian for the perspective projection (2.1) is:

$$\mathbf{J}_{\mathbf{w}} = \begin{pmatrix} \frac{fx}{z} & \frac{fy}{z} & f & \frac{f}{z} & 0 & 0 & 0 & 0 & -\frac{fx^2}{z^2} & -\frac{fxy}{z^2} & -\frac{fx}{z} & -\frac{fx}{z^2} \\ 0 & 0 & 0 & 0 & \frac{fx}{z} & \frac{fy}{z} & f & \frac{f}{z} & -\frac{fxy}{z^2} & -\frac{fy^2}{z^2} & -\frac{fy}{z} & -\frac{fy}{z^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{A.4}$$

with f the focal length. This Jacobian for the spherical projection model using eq. (2.6) is:

$$\mathbf{J_w} = \begin{pmatrix} -\frac{xz}{x^2+z^2} & -\frac{yz}{x^2+z^2} & -\frac{z^2}{x^2+z^2} & -\frac{z}{x^2+z^2} & 0 & 0 \\ -\frac{x^2y}{\rho^2\sqrt{x^2+z^2}} & -\frac{xy^2}{\rho^2\sqrt{x^2+z^2}} & -\frac{xyz}{\rho^2\sqrt{x^2+z^2}} & -\frac{xy}{\rho^2\sqrt{x^2+z^2}} & \frac{x\sqrt{x^2+z^2}}{\rho^2} & \frac{y\sqrt{x^2+z^2}}{\rho^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{x^2}{x^2+z^2} & \frac{xy}{x^2+z^2} & \frac{xz}{x^2+z^2} & \frac{x}{x^2+z^2} \\ \frac{z\sqrt{x^2+z^2}}{\rho^2} & \frac{\sqrt{x^2+z^2}}{\rho^2} & -\frac{xyz}{\rho^2\sqrt{x^2+z^2}} & -\frac{y^2z}{\rho^2\sqrt{x^2+z^2}} & -\frac{yz^2}{\rho^2\sqrt{x^2+z^2}} & -\frac{yz}{\rho^2\sqrt{x^2+z^2}} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{A.5}$$

where $\rho = \sqrt{x^2+y^2+z^2}$.

A.2 Geometric Error Jacobians

The geometric error is the point-to-plane ICP error:

$$e_D(\mathbf{p},\mathbf{x}) = \lambda_D(\hat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{n}^*(\mathbf{p}))^T \left(\mathbf{P}(w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) - \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}) \begin{pmatrix} \mathbf{P}^*(\mathbf{p}) \\ 1 \end{pmatrix}\right). \tag{A.6}$$

For simplicity of notation to compute the Jacobian $\mathbf{J}^D \in \mathbb{R}^{1\times 6}$, we denote the 3D point error $\zeta(\mathbf{x})$:

$$\begin{aligned} \zeta(\mathbf{x}) &= -\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}) \begin{pmatrix} \mathbf{P}^*(\mathbf{p}) \\ 1 \end{pmatrix} + \mathbf{P}(w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) \\ &= -\hat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{P}^*(\mathbf{p}) - \hat{\mathbf{R}}\mathbf{t}(\mathbf{x}) - \hat{\mathbf{t}} + \mathbf{P}(w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))). \end{aligned} \tag{A.7}$$

And therefore eq. (A.6) becomes:

$$e_D(\mathbf{p},\mathbf{x}) = \lambda_D(\hat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{n}^*(\mathbf{p}))^T\zeta(\mathbf{x}). \tag{A.8}$$

From eqs. (A.8), (A.7) and the product rule:

$$\mathbf{J}^D(\mathbf{0}) = \lambda_D\mathbf{n}^{*T}\left(\left.\frac{\partial(\mathbf{R}(\mathbf{x})^T\hat{\mathbf{R}}^T\zeta(\mathbf{z}))}{\partial\mathbf{x}}\right|_{\mathbf{z}=\mathbf{x}} + \mathbf{R}(\mathbf{x})^T\hat{\mathbf{R}}^T\nabla_{\mathbf{x}}(\zeta(\mathbf{x}))\right)\Bigg|_{\mathbf{x}=\mathbf{0}}. \tag{A.9}$$

For clarity, the first term in eq. (A.9) is $\mathbf{J_{d1}}$ and we decompose the second term in two Jacobians $\mathbf{J_{d2}}$ and $\mathbf{J_{d3}}$, such as $\mathbf{J}^D(\mathbf{0}) = \lambda\mathbf{n}^{*T}(\mathbf{J_{d1}}(\mathbf{0}) + \mathbf{J_{d2}}(\mathbf{0}) + \mathbf{J_{d3}}(\mathbf{0}))$. From $\frac{\partial(\mathbf{R}(\mathbf{x})\zeta)}{\partial\mathbf{x}} =$

$\frac{\partial(\mathbf{R}(\mathbf{x})\zeta)}{\partial\mathbf{R}(\mathbf{x})}\frac{\partial\mathbf{R}(\mathbf{x})}{\partial\mathbf{x}}$, the first term is:

$$\mathbf{J_{d1}}(\mathbf{0}) = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{S}(\hat{\mathbf{R}}^{\mathbf{T}}\zeta(\mathbf{0})) \end{pmatrix}. \tag{A.10}$$

The second term is decomposed in two Jacobians:

$$\mathbf{J_{d2}}(\mathbf{0}) = \begin{pmatrix} -\mathbf{I}_{3\times3} & \mathbf{S}(\mathbf{P}^*(\mathbf{p})) \end{pmatrix}. \tag{A.11}$$

And finally the last Jacobian is the one corresponding to $\nabla_{\mathbf{x}}(\mathbf{P}(w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})))$. This Jacobian can be seen as an extended version of the image photometric gradient $\mathbf{J^I}$, for each component of $\mathbf{P}(w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}))$:

$$\mathbf{J_{d3}}(\mathbf{0}) = \begin{pmatrix} \mathbf{J_P}\big|^T_{[\mathbf{P}(\mathbf{p}_w)]_1} & \mathbf{J_P}\big|^T_{[\mathbf{P}(\mathbf{p}_w)]_2} & \mathbf{J_P}\big|^T_{[\mathbf{P}(\mathbf{p}_w)]_3} \end{pmatrix}^T \mathbf{J_w}\mathbf{J_T}, \tag{A.12}$$

where $\mathbf{J_P}\big|_{[\mathbf{P}(\mathbf{p}_w)]_i}$ is the image gradient (as in the photometric term) of an image produced with the ith-coordinate of $\mathbf{P}(w(\mathbf{p},\hat{\mathbf{T}}\mathbf{T}(\mathbf{0})))$. Note that this Jacobian is small for points belonging to planar surfaces and have a high computation effort. Therefore, $\mathbf{J_{d3}}$ is neglected since only a fraction of the scene is on geometric discontinuities and since these points have higher sensitivity to depth error estimates and self-occlusions.

A.3 Normal Vector Error Jacobians

Similarly to the direct image registration framework, the errors using the normals can be modeled as:

$$\mathbf{e}_{N1}(\mathbf{p},\boldsymbol{\omega}) = \mathbf{n}(w(\mathbf{p},\hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))) - \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}) \tag{A.13}$$

$$\mathbf{e}_{N2}(\mathbf{p},\boldsymbol{\omega}) = \mathbf{n}(w(\mathbf{p},\hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))) \times \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}) \tag{A.14}$$

where $\hat{\mathbf{R}}$ is a first rotation approximation. For the normal error metric (A.14), $\mathbf{J}^{N2} \in \mathbb{R}^{3\times3}$ is:

$$\nabla_{\boldsymbol{\omega}}(\mathbf{e}_{N2}) = \nabla_{\boldsymbol{\omega}}\mathbf{n}(w(\mathbf{p},\hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))) \times \hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}) + \mathbf{n}(w(\mathbf{p},\hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))) \times \hat{\mathbf{R}}\frac{\partial(\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}))}{\partial\boldsymbol{\omega}} \tag{A.15}$$

$$\mathbf{J}^{N2}(\mathbf{0}) = -\mathbf{S}(\hat{\mathbf{R}}\mathbf{n}^*(\mathbf{p}))\nabla_{\boldsymbol{\omega}}(\mathbf{n}(w(\mathbf{p},\hat{\mathbf{R}}\mathbf{R}(\boldsymbol{\omega}))))\Big|_{\boldsymbol{\omega}=\mathbf{0}} + \mathbf{S}(\mathbf{n}(w(\mathbf{p},\hat{\mathbf{R}})))\hat{\mathbf{R}}\frac{\partial(\mathbf{R}(\boldsymbol{\omega})\mathbf{n}^*(\mathbf{p}))}{\partial\boldsymbol{\omega}}\Big|_{\boldsymbol{\omega}=\mathbf{0}}. \tag{A.16}$$

The first term of (A.15) is analogous to the Jacobian in eq. (A.12) and as so it is zero in all points belonging to planar surfaces. We will neglect this term for computational reasons since only a fraction of the scene is on geometric frontiers and these points have higher sensitivity to depth error estimates and self-occlusions effects. Then eq. (A.16) yields to:

$$\mathbf{J}^{N2}(\mathbf{0}) = -\mathbf{S}(\mathbf{n}(w(\mathbf{p}, \hat{\mathbf{R}})))\hat{\mathbf{R}}\mathbf{S}(\mathbf{n}^*(\mathbf{p})) \quad \in \mathbb{R}^{(3\times 3)}. \tag{A.17}$$

Note that (A.17) depends on the normal vectors in the current and reference frames. By last, the Jacobian of (A.13) is simply:

$$\mathbf{J}^{N1}(\mathbf{0}) = -\hat{\mathbf{R}}\mathbf{S}(\mathbf{n}^*(\mathbf{p})) \in \mathbb{R}^{(3\times 3)} \tag{A.18}$$

which does not depend of the warped current normal vectors. The simplification used in eqs. (A.12) and (A.15) is also applied to this Jacobian.

A.4 Robust M-Estimators

Outliers are errors that cannot be explained by the model underlying the intensity and geometric error distributions, such as in the presence of occlusions, moving objects, wrong depth estimation or changes of illumination. These outliers can have a strong effect in the minimization of the cost function. Robust M-Estimators are a class of estimation methods that can handle outliers present in the sensor data, by replacing the Euclidean norm of the residuals (which assumes an error with Gaussian/Normal distribution) by a less increasing or even a bounded norm. We follow the treatment given in section 9.4 of [Zhang, 1995] for these estimators. Instead of using the Euclidean norm, we want to minimize the cost:

$$C = \min_{\mathbf{x}} \sum_{\mathbf{p}} \rho(\mathbf{e}(\mathbf{p}, \mathbf{x})), \tag{A.19}$$

where ρ is a symmetric positive distance chosen to be less increasing than the Euclidean norm. This problem can be solved iteratively using a re-weighted least squares system:

$$C = \min_{\mathbf{x}} \sum_{\mathbf{p}} w_i(\mathbf{e}_{k-1}(\mathbf{p}, \mathbf{x}))\|\mathbf{e}_k(\mathbf{p}, \mathbf{x})\|_2^2, \tag{A.20}$$

where the weight $w_i(x) = \Phi(x)/x$ and $\Phi(x) = d(\rho(x))/dx$ is called the influence function. Some commonly used robust distances and the respective influence functions are shown in fig. A.1. Before computing the weights, we centralize and re-scale the errors using the median and robust

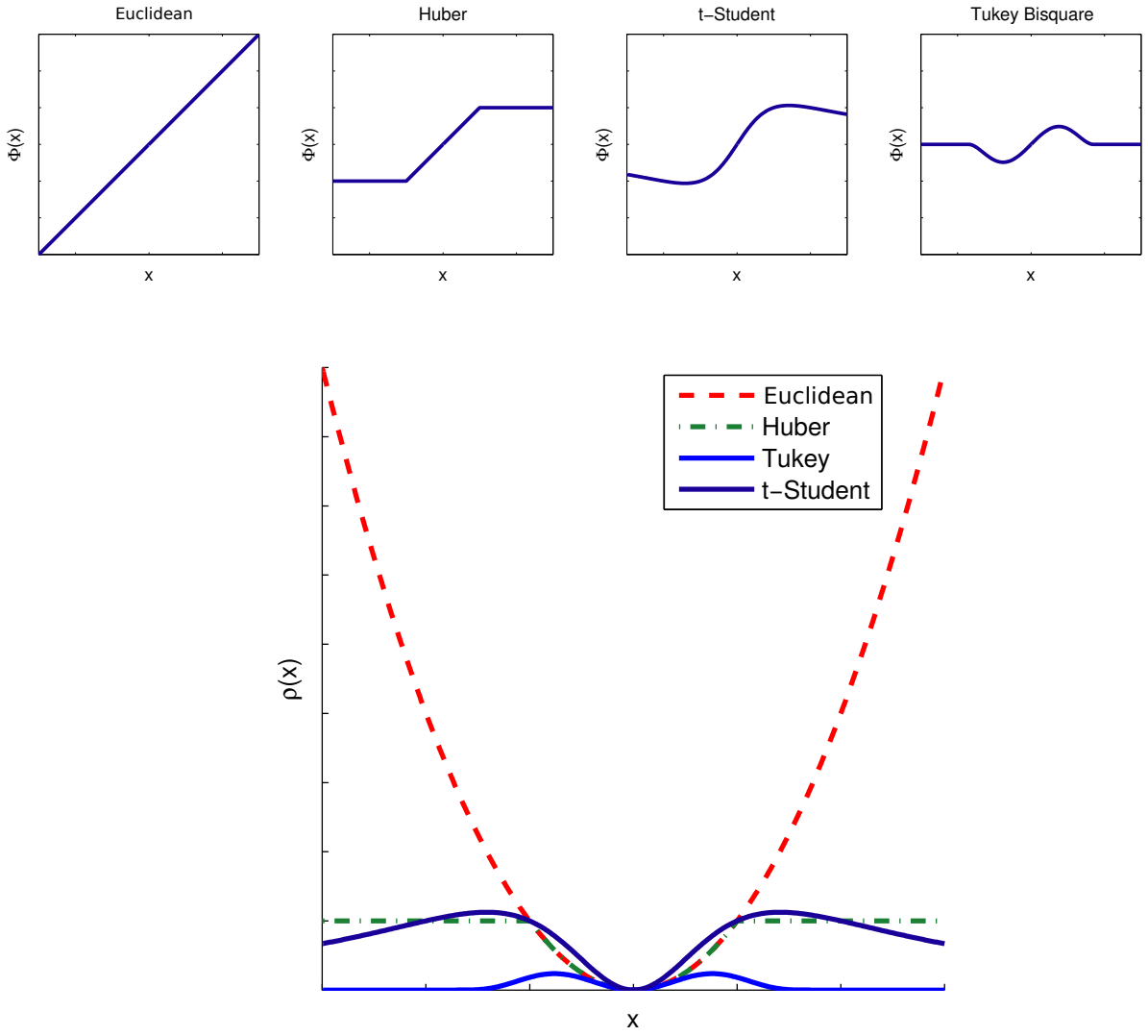

Figure A.1 – Influence and robust functions of commonly used M-Estimators: the Euclidean norm, Huber, t-student and Tukey bi-square.

variance:

$$x(\mathbf{p}) = \frac{(\mathbf{e}(\mathbf{p}) - \text{median}(\mathbf{e}))}{\hat{\sigma}}, \tag{A.21}$$

where $\hat{\sigma}$ is a robust estimate of the covariance using the MAD:

$$\hat{\sigma} = \frac{1}{\Psi^{-1}(0.75)} \text{median}(\|\delta(\mathbf{p}) - \text{median}(\delta))\|_1). \tag{A.22}$$

with $\delta(\mathbf{p}) = \mathbf{e}(\mathbf{p}) - \text{median}(\mathbf{e})$ and Ψ is the cumulative normal distribution function, where $\Psi^{-1}(0.75) = 1.48$ represents one standard deviation of the normal distribution.

Appendix B

Error Propagation and Keyframe Fusion

In order to fuse the information of different RGB-D-C frames, captured from different viewpoints, the frames and their uncertainty matrices need to be warped to the same reference pose of the keyframe. Warping the augmented frame $\boldsymbol{\mathcal{S}}$ generates a synthetic view of the scene $\boldsymbol{\mathcal{S}}_w = \{\boldsymbol{\mathcal{I}}_w, \boldsymbol{\mathcal{D}}_w, \Sigma_{\mathcal{D}_w}\}$, as it would appear from a new viewpoint. The intensity and depth images in a viewpoint related by the pose $\mathbf{T}$ are:

$$\begin{aligned} \boldsymbol{\mathcal{I}}_w(w(\mathbf{p}, \mathbf{T})) &= \boldsymbol{\mathcal{I}}(\mathbf{p}), \\ \boldsymbol{\mathcal{D}}_w(w(\mathbf{p}, \mathbf{T})) &= \|\mathbf{R}\Pi^{-1}(\mathbf{p})\boldsymbol{\mathcal{D}}(\mathbf{p}) + \mathbf{t}\|_2, \end{aligned} \tag{B.1}$$

assuming stationary photometric and geometric models, as described in section 3.3.1 of chapter 3 and with minimal occlusions between the different viewpoints.

B.1 Uncertainty Propagation

The confidence of the elements in $\boldsymbol{\mathcal{D}}_w$ clearly depends on the combination of the pixel position, the depth and the pose errors over a set of geometric and projective operations. The propagation assumes the noise present in the images as Gaussian because they can be represented by the two first moments and by the property that any linear combination of Gaussian distributions is Gaussian. Herein we describe the propagation for the spherical sensor. For notation compactness, we omit the pixel coordinates and we refer to the unit vector in the direction of the pixel $\mathbf{p}$ as $\mathbf{q} = \Pi_S^{-1}(\mathbf{p})$. We start by propagating the uncertainty of the depth and pixel location to the Cartesian 3D point $\mathbf{P}(\mathbf{p})$. Considering the spherical projection model, taking a first order approximation of $\mathbf{P}(\mathbf{p}) = \boldsymbol{\mathcal{D}}(\mathbf{p})\Pi_S^{-1}(\mathbf{p})$, the error covariance can be decomposed as:

$$\Sigma_{\mathbf{P}} = \sigma_{\boldsymbol{\mathcal{D}}}^2 \mathbf{q}\mathbf{q}^T + \boldsymbol{\mathcal{D}}^2 \Sigma_{\mathbf{q}} \tag{B.2}$$

where σ_D^2 is the depth uncertainty from the regularization of chapter 6 and $\Sigma_{\mathbf{q}}$ is the pixel coordinate confidence (in general $\Sigma_{\mathbf{q}}$ is small). The next step consists of combining an uncertain rigid transform $\mathbf{T}$ with the uncertainty in $\mathbf{P}$. Given the mean of the 6DOF $\bar{\mathbf{y}} = \{t_x, t_y, t_z, \theta, \varphi, \psi\}$ in 3D+YPR form and its covariance $\Sigma_{\mathbf{y}}$, then for $\mathbf{P}_w(\mathbf{p}, \mathbf{T}) = \mathbf{R}\mathbf{P}(\mathbf{p}) + \mathbf{t}$ we have:

$$\begin{aligned} \Sigma_{\mathbf{P}_w} &= \mathbf{J}_{\mathbf{P}}(\mathbf{P}, \bar{\mathbf{y}})\boldsymbol{\Sigma}_{\mathbf{P}}\mathbf{J}_{\mathbf{P}}(\mathbf{P}, \bar{\mathbf{y}})^{\mathbf{T}} + \mathbf{J}_{\mathbf{T}}(\mathbf{P}, \bar{\mathbf{y}})\boldsymbol{\Sigma}_{\mathbf{y}}\mathbf{J}_{\mathbf{T}}(\mathbf{P}, \bar{\mathbf{y}})^{\mathbf{T}} \\ &= \mathbf{R}\Sigma_{\mathbf{P}}\mathbf{R}^T + \mathbf{M}\Sigma_{\mathbf{T}}\mathbf{M}^T \end{aligned} \tag{B.3}$$

Where $\Sigma_{\mathbf{P}}$ is as in (B.2) and $\mathbf{M} \approx \begin{bmatrix} \mathbf{I}_{(3\times 3)} & \begin{matrix} -y & z & 0 \\ x & 0 & -z \\ 0 & -x & y \end{matrix} \end{bmatrix}$ for small rotations (see [Blanco, 2010] for the general formula of $\mathbf{M}$). The depth image warped in the pose $\mathbf{T}$ is as in eq. (B.1): $\boldsymbol{\mathcal{D}}_w = \|\mathbf{P}_w\|_2$ and thus, the covariance represented in the reference pose coordinate system is:

$$\sigma^2_{\mathcal{D}_w} = \mathbf{J}_S(\mathbf{P}_w)\Sigma_{\mathbf{P}_w}\mathbf{J}_S(\mathbf{P}_w)^T \tag{B.4}$$

where $\mathbf{J}_S$ is the Jacobian of the Euclidean norm: $\mathbf{J}_S(\mathbf{z}) = (\mathbf{z}^{\mathbf{T}}/\sqrt{\mathbf{z}^{\mathbf{T}}\mathbf{z}})$.

Bibliography

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixels methods. *IEEE Trans. PAMI*, 34(11). [6.2, 6.4.2]

[Alismail et al., 2016] Alismail, H., Browning, B., and Lucey, S. (2016). Robust tracking in low light and sudden illumination changes. In *3DV*. [3.2.2]

[Anguelov et al., 2010] Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., and Weaver, J. (2010). Google street view: Capturing the world at street level. *Computer*, 43(6). [2.1, 2.1, 2.2, 7.1, 7.2]

[Arun et al., 1987] Arun, K., Huang, T., and Blostein, S. (1987). Least-squares fitting of two 3-D point sets. *IEEE Trans. on PAMI*, 19(5). [4.1.1]

[Bachrach et al., 2012] Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A. S., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2012). Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. *IJRR*, 31(11). [7.2]

[Badino et al., 2011a] Badino, H., Huber, D., Park, Y., and Kanade, T. (2011a). Fast and accurate computation of surface normals from range images. In *IEEE ICRA*. [2.5, 2.5.1.2]

[Badino et al., 2011b] Badino, H., Huber, D., Park, Y., and Kanade, T. (2011b). Fast and accurate computation of surface normals from range images. In *IEEE ICRA*. [4.1.3.1]

[Baker and Matthews, 2001] Baker, S. and Matthews, I. (2001). Equivalence and efficiency of image alignment algorithms. In *IEEE CVPR*. [3.2.2, 3.3.2]

[Baker and Matthews, 2006] Baker, S. and Matthews, I. (2006). Lucas-Kanade 20 years on: a unifying framework. *IJCV*, 56. [3.2.2, 3.3.1, 3.3.3.2]

[Barreto, 2006] Barreto, J. (2006). A unifying geometric representation for central projection systems. *CVIU*, 103(3). [2.2]

[Benhimane and Malis, 2004] Benhimane, S. and Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE IROS*. [3.3.3.2]

[Berenguer et al., 2015] Berenguer, Y., Payá, L., Ballesta, M., and Reinoso, Ó. (2015). Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors*, 15(10). [4.1.1]

[Blanco, 2010] Blanco, J. (2010). A tutorial on SE(3) transformation parameterizations and on-manifold optimization. Technical Report 012010, U. Malaga. [A.1, B.1]

[Braux-Zin et al., 2013] Braux-Zin, J., Dupont, R., and Bartoli, A. (2013). A general dense image matching framework combining direct and feature-based costs. In *IEEE ICCV*. [3.4, 5.1.1]

[Brox and Malik, 2011] Brox, T. and Malik, J. (2011). Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE PAMI*, 33. [3.4, 5.1.1]

[Buczko and Willert, 2016] Buczko, M. and Willert, V. (2016). Flow-decoupled normalized reprojection error for visual odometry. In *IEEE ITSC*. [3.2.1]

[Bulow, 2002] Bulow, T. (2002). Multiscale image processing on the sphere. In *DAGM Symposium on Pattern Recognition*. [2.4, 2.4, 2.6]

[Burt and Adelson, 1987] Burt, P. and Adelson, E. (1987). Readings in computer vision: Issues, problems, principles, and paradigms. chapter The Laplacian Pyramid As a Compact Image Code. [5.2]

[Calakli and Taubin, 2011] Calakli, F. and Taubin, G. (2011). SSD: Smooth signed distance surface reconstruction. *Comput. Graph. Forum*, 30(7). [7.2]

[Calonder et al., 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary robust independent elementary features. In *IEEE ECCV*. [3.2.1]

[Caron et al., 2011] Caron, G., Marchand, E., and Mouaddib, E. (2011). Tracking planes in omnidirectional stereovision. In *IEEE ICRA*. [3.2.2]

[Chambolle, 2004] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2). [5.1.1, 6.2]

[Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1). [6.2]

[Chapoulie, 2012] Chapoulie, A. (2012). *Contributions to visual loop closure detection and environment topological segmentation methods.* PhD thesis, Univ. Nice Sophia Antipolis. [2.3.3]

[Chapoulie et al., 2011] Chapoulie, A., Rives, P., and Filliat, D. (2011). A spherical representation for efficient visual loop closing. In *IEEE OMNIVIS*. [2.1, 7.1]

[Chaumette and Hutchinson, 2006] Chaumette, F. and Hutchinson, S. (2006). Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4). [5.2.1]

[Chiu et al., 2016] Chiu, H., Sizintsev, M., Zhou, X., Miller, P., Samarasekera, S., and Kumar, R. (2016). Sub-meter vehicle navigation using efficient pre-mapped visual landmarks. In *IEEE ITSC*. [7.1, 7.2]

[Churchill et al., 2015] Churchill, W., Tong, C., Gurau, C., Posner, I., and Newman, P. (2015). Know your limits: Embedding localiser performance models in teach and repeat maps. In *IEEE ICRA*. [3.4, 3.6, 5.5, 8.1, 1]

[Civera et al., 2008] Civera, J., Davison, A., and Montiel, J. (2008). Inverse depth parametrization for monocular SLAM. *IEEE Trans. Robotics*, 24(5). [6.4.4]

[Comport et al., 2007] Comport, A., Malis, E., and Rives, P. (2007). Accurate quadrifocal tracking for robust 3D visual odometry. In *IEEE ICRA*. [3.2.2]

[Comport et al., 2010] Comport, A., Malis, E., and Rives, P. (2010). Real-time quadrifocal visual odometry. *IJRR*, 29. [3.2, 3.1, 3.2.2, 3.2.3, 3.3.2, 3.4, 5.1]

[Corke and Mahony, 2009] Corke, P. and Mahony, R. (2009). Sensing and control on the sphere. In *Robotics Research : The 14th International Symposium ISSR*. [4.2.2, 8.1, 1]

[Costante et al., 2016] Costante, G., Forster, C., Delmerico, J., Valigi, P., and Scaramuzza, D. (2016). Perception-aware path planning. *CoRR*, abs/1605.04151. [8.1, 1]

[Dame and Marchand, 2010] Dame, A. and Marchand, E. (2010). Accurate real-time tracking using mutual information. In *IEEE ISMAR*. [3.2.2]

[Daniilidis et al., 2002] Daniilidis, K., Makadia, A., and Bulow, T. (2002). Image processing in catadioptric planes: Spatiotemporal derivatives and optical flow computation. In *IEEE OMNIVIS*. [2.4, 2.6]

[Das and Waslander, 2015] Das, A. and Waslander, S. (2015). Entropy based keyframe selection for multi-camera visual SLAM. In *IEEE IROS*. [7.2]

[Davison and Murray, 2002] Davison, A. and Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Trans. on PAMI*, 24(7). [3.2.1]

[Dayoub et al., 2011] Dayoub, F., Cielniak, G., and Duckett, T. (2011). Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *RAS*, 59(5). [2.1, 7.1, 7.2, 7.3]

[Dellaert, 2012] Dellaert, F. (2012). Factor graphs and gtsam: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM. [7.4]

[Demonceaux et al., 2011] Demonceaux, C., Vasseur, P., and Fougerolle, Y. (2011). Central catadioptric image processing with geodesic metric. *Image Vision Comput.*, 29(12). [2.4, 2.4, 2.6]

[Dequaire et al., 2016] Dequaire, J., Tong, C., Churchill, W., and Posner, I. (2016). Off the beaten track: Predicting localisation performance in visual teach and repeat. In *IEEE ICRA*. [3.4, 3.6, 5.5, 8.1, 1]

[Dosovitskiy et al., 2015] Dosovitskiy, A., Fischer, P., Ilg, E., Haeusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *IEEE ICCV*. [5, 8.1, 1]

[Drouilly, 2015] Drouilly, R. (2015). *Cartographie hybride métrique topologique et sémantique pour la navigation dans de grands environnements*. PhD thesis, Univ. of Nice. [2.3.3]

[Dryanovski et al., 2013] Dryanovski, I., Valenti, R., and Xiao, J. (2013). Fast visual odometry and mapping from RGB-D data. In *IEEE ICRA*. [3.2.1, 3.2.3, 3.2.3, 6.4.4, 7.2]

[Duan and Lafarge, 2015] Duan, L. and Lafarge, F. (2015). Image partitioning into convex polygons. In *IEEE CVPR*. [6.2]

[Edelsbrunner, 2000] Edelsbrunner, H. (2000). Triangulations and meshes in computational geometry. *Acta Numerica*. [6.2]

[Engel et al., 2015] Engel, J., Stückler, J., and Cremers, D. (2015). Large-scale direct SLAM with stereo cameras. In *IEEE IROS*. [3.1, 3.2]

[Ester et al., 1996] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*. [6.4.3]

[Faugeras et al., 2001] Faugeras, O., Luong, Q., and Papadopoulou, T. (2001). *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press. [2.2, 2.3.2, 3.1]

[Fernandez-Moral, 2014] Fernandez-Moral, E. (2014). *Contributions to metric-topological localization and mapping in mobile robotics*. PhD thesis, Univ. of Malaga. [2.3.3]

[Fernandez-Moral et al., 2014] Fernandez-Moral, E., Gonzalez-Jimenez, J., Rives, P., and Arevalo, V. (2014). Extrinsic calibration of a set of range cameras in 5 seconds without pattern. In *IEEE IROS*. [2.2, 2.3.1, 2.3.3, 4.1.1, 4.2.1.4, 5.4.3]

[Fernandez-Moral et al., 2017] Fernandez-Moral, E., Martins, R., Wolf, D., and Rives, P. (2017). A new metric for evaluating semantic segmentation: Leveraging global and contour accuracy. In *IEEE IROS PPNIV*. [3.1, 1.3.1]

[Fernandez-Moral et al., 2013] Fernandez-Moral, E., Mayol-Cuevas, W., Arevalo, V., and Gonzalez-Jimenez, J. (2013). Fast place recognition with plane-based maps. In *IEEE ICRA*. [6.1, 7.2]

[Fernandez-Moral et al., 2016] Fernandez-Moral, E., Rives, P., Arevalo, V., and Gonzalez-Jimenez, J. (2016). Scene structure registration for localization and mapping. *RAS*, 75(Part B). [7.2]

[Florez et al., 2012] Florez, S. R., Frémont, V., Bonnifait, P., and Cherfaoui, V. (2012). An embedded multi-modal system for object localization and tracking. *IEEE Intell. Transport. Syst. Mag.*, 4(4). [3.2.2]

[Furgale and Barfoot, 2010] Furgale, P. and Barfoot, T. (2010). Visual teach and repeat for long-range rover autonomy. *JFR*, 27. [3.4]

[Gallier, 2000] Gallier, J. (2000). *Geometric Methods and Applications For Computer Science and Engineering*. [3.3.3.1, 3.3.3.1]

[Gallier and Xu, 2002] Gallier, J. and Xu, D. (2002). Computing exponentials of skew symmetric matrices and logarithms of orthogonal matrices. *International Journal of Robotics and Automation*, 17(4). [3.3.3.1]

[Gaspar et al., 2000] Gaspar, J., Winters, N., and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Trans. on Rob. and Automation*, 16(6). [2.1]

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE CVPR*. [5.1, 5.3.2]

[Geiger et al., 2010] Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient large-scale stereo matching. In *ACCV*. [2.3.2, 5.4.4, 6.3.1]

[Gelfand et al., 2003] Gelfand, N., Ikemoto, L., Rusinkiewicz, S., and Levoy, M. (2003). Geometrically stable sampling for the ICP algorithm. In *3DIM*. [3.3.2, 4.2.2.1, 4.3, 5.1]

[Geyer and Daniilidis, 2001] Geyer, C. and Daniilidis, K. (2001). Catadioptric projective geometry. *IJCV*, 45(3). [2.2]

[Gluckman et al., 1998] Gluckman, J., Nayar, S., and Thoresz, K. (1998). Real-time omnidirectional and panoramic stereo. In *DARPA Image Understanding Workshop*. [2.3.2]

[Gokhool, 2015] Gokhool, T. (2015). *A Compact RGB-D Map Representation dedicated to Autonomous Navigation*. PhD thesis, Univ. of Nice. [2.3.3]

[Gokhool et al., 2015] Gokhool, T., Martins, R., Rives, P., and Despre, N. (2015). A compact spherical RGBD keyframe-based representation. In *IEEE ICRA*. [3.1, 1.3.1, 3.2.3, 5.1, 7.2, 7.3, 7.4]

[Guney and Geiger, 2016] Guney, F. and Geiger, A. (2016). Deep discrete flow. In *ACCV*. [8.1, 1]

[Gutierrez-Gomez et al., 2016] Gutierrez-Gomez, D., Mayol-Cuevas, W., and Guerrero, J. (2016). Dense RGB-D visual odometry using inverse depth. *Robot. Auton. Syst.*, 75(PB). [3.2.3, 6.4.4]

[Hadj-Abdelkader et al., 2008] Hadj-Abdelkader, H., Malis, E., and Rives, P. (2008). Spherical image processing for accurate visual odometry with omnidirectional cameras. In *IEEE OMNIVIS*. [2.4, 2.4, 2.6, 3.4, 4.1, 5.1.1]

[Hager and Belhumeur, 1998] Hager, G. and Belhumeur, P. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI*, 20(10). [3.2.2]

[Hamann, 1993] Hamann, B. (1993). Curvature approximation for triangulated surfaces. *Computing*, 8. [2.5]

[Han, 2013] Han, D. (2013). Comparison of commonly used image interpolation methods. In *ICCSEE*. [3.3.1.1]

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *4th Alvey Vision Conf.* [3.2.1]

[Hartley, 1997] Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Trans. on PAMI*, 19(6). [3.2.1]

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press. [2.2, 2.3.2, 3.1, 3.3.3.2, 6.4.3]

[Heng and Choi, 2016] Heng, L. and Choi, B. (2016). Semi-direct visual odometry for a fisheye-stereo camera. In *IEEE IROS*. [3.2, 3.3.2]

[Henry et al., 2014] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2014). *RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments*. [3.2.3, 6.1, 7.2]

[Hirschmuller, 2008] Hirschmuller, H. (2008). Stereo processing by semi-global matching and mutual information. *IEEE Trans. PAMI*, 30(2). [2.3.2, 5.1.1, 6.2, 6.3.1]

[Hoppe et al., 1992] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. In *ACM SIGGRAPH*. [2.5]

[Horn and Schunck, 1981] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1). [3.2.1, 5]

[Huhle et al., 2010] Huhle, B., Schairer, T., Jenke, P., and Straßer, W. (2010). Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *CVIU*, 114. [6.2, 7.1]

[Irani and Anandan, 2000] Irani, M. and Anandan, P. (2000). About direct methods. In *IEEE ICCV Workshop on Vision Algorithms: Theory and Practice*. [3.2.2]

[Jaimez et al., 2017] Jaimez, M., Kerl, C., Gonzalez-Jimenez, J., and Cremers, D. (2017). Fast odometry and scene flow from RGB-D cameras based on geometric clustering. In *IEEE ICRA*. [5, 3.2.3]

[Jogan and Leonardis, 2000] Jogan, M. and Leonardis, A. (2000). Robust localization using panoramic view-based recognition. In *ICPR*, volume 4. [2.1]

[Jordan and Mordohai, 2014] Jordan, K. and Mordohai, P. (2014). A quantitative evaluation of surface normal estimation in point clouds. In *IEEE IROS*. [2.5]

[Kelly and Sukhatme, 2011] Kelly, J. and Sukhatme, G. (2011). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *IJRR*, 30(1). [4.1]

[Kendall et al., 2015] Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *IEEE ICCV*. [8.1, 1]

[Kerl et al., 2015] Kerl, C., Stuckler, J., and Cremers, D. (2015). Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras. In *IEEE ICCV*. [3.2.3, 3.3, 1, 5.1.1, 5.4]

[Kerl et al., 2013a] Kerl, C., Sturm, J., and Cremers, D. (2013a). Dense visual SLAM for RGB-D cameras. In *IEEE IROS*. [3.2.3, 3.2.3, 3.3.2]

[Kerl et al., 2013b] Kerl, C., Sturm, J., and Cremers, D. (2013b). Dense visual SLAM for RGB-D cameras. In *IEEE IROS*. [5.1, 5.1.1, 7.2]

[Khoshelham and Elberink, 2012] Khoshelham, K. and Elberink, S. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2). [3.2.3, 6.4.4]

[Kim and Chung, 2003] Kim, J. and Chung, M. (2003). SLAM with omni-directional stereo vision sensor. In *IEEE IROS*, volume 1. [2.1]

[Kitt et al., 2010] Kitt, B., Geiger, A., and Lategahn, H. (2010). Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *IEEE IV*. [3.2.1]

[Klasing et al., 2009] Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *IEEE ICRA*. [2.5, 4.1.3.1]

[Klose et al., 2013] Klose, S., Heise, P., and Knoll, A. (2013). Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data. In *IEEE IROS*. [3.2.2, 3.4]

[Konda and Memisevic, 2015] Konda, K. and Memisevic, R. (2015). Learning visual odometry with a convolutional network. In *VISAPP*. [8.1, 1]

[Korn et al., 2014] Korn, M., Holzkothen, M., and Pauli, J. (2014). Color supported generalized-ICP. In *VISAPP*. [3.2.3, 3.3.2, 5.1.1]

[Lébraly et al., 2010] Lébraly, P., Royer, E., Ait-Aider, O., and Dhome, M. (2010). Calibration of non-overlapping cameras—application to vision-based robotics. In *BMVC*. [2.3.1]

[Li and Hartley, 2007] Li, H. and Hartley, R. (2007). The 3D-3D registration problem revisited. In *IEEE ICCV*. [4.1.1, 3]

[Lim and Barnes, 2008] Lim, J. and Barnes, N. (2008). Directions of egomotion from antipodal points. In *IEEE CVPR*. [8.1, 1]

[Liu et al., 2014] Liu, Y., Li, Y., Lou, J., Wang, W., and Zhang, M. (2014). Omni-total variation algorithm for the restoration of all-focused catadioptric image. *Optik - IJLEO*, 125(14). [2.1]

[Lowe, 2004] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2). [3.2.1]

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI*. [3.2.2]

[Ma et al., 2016] Ma, Y., Guo, Y., Zhao, J., Lu, M., Zhang, J., and Wan, J. (2016). Fast and accurate registration of structured point clouds with small overlaps. In *IEEE CVPR Workshops*. [4.1, 4.1.1, 4.1.2, 4.2.1.2, 4.2.1.3]

[Maier et al., 2015] Maier, R., Stueckler, J., and Cremers, D. (2015). Super-resolution keyframe fusion for 3D modeling with high-quality textures. In *3DV*. [5.1, 7.2]

[Maimone et al., 2007] Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the Mars exploration rovers. *Journal of Field Robotics*, 24(3). [3.2, 4.1]

[Malis and Vargas, 2007] Malis, E. and Vargas, M. (2007). Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA. [2, 3.2.2]

[Marchand et al., 2016] Marchand, E., Uchiyama, H., and Spindler, F. (2016). Pose estimation for augmented reality: A hands-on survey. *IEEE Trans. Vis. Comput. Graph.*, 22(12). [3.1]

[Martins et al., 2015] Martins, R., Fernandez-Moral, E., and Rives, P. (2015). Dense accurate urban mapping from spherical RGB-D images. In *IEEE IROS*. [3.1, 1.3.1, 6.4.1]

[Martins et al., 2016] Martins, R., Fernandez-Moral, E., and Rives, P. (2016). Adaptive direct RGB-D registration and mapping for large motions. In *ACCV*. [3.1, 1.3.1]

[Martins et al., 2017] Martins, R., Fernandez-Moral, E., and Rives, P. (2017). An efficient rotation and translation decoupled initialization from large field of view depth images. In *IEEE IROS*. [3.1, 1.3.1]

[Martins and Rives, 2016] Martins, R. and Rives, P. (2016). Increasing the convergence domain of RGB-D direct registration methods for vision-based localization in large scale environments. In *IEEE ITSC PPNIV*. [3.1, 1.3.1]

[Mei et al., 2006] Mei, C., Benhimane, S., Malis, E., and Rives, P. (2006). Constrained multiple planar template tracking for central catadioptric cameras. In *BMVC*. [3.2.2]

[Mei and Rives, 2006] Mei, C. and Rives, P. (2006). Calibration between a central catadioptric camera and a laser range finder for robotics applications. In *IEEE ICRA*. [2.3.1]

[Mei and Rives, 2007] Mei, C. and Rives, P. (2007). Single view point omnidirectional camera calibration from planar grids. In *IEEE ICRA*. [2.2]

[Meilland, 2012] Meilland, M. (2012). *Dense RGB-D mapping for real-time localisation and autonomous navigation*. PhD thesis, MINES ParisTech. [2.3.3]

[Meilland and Comport, 2013] Meilland, M. and Comport, A. (2013). On unifying key-frame and voxel-based dense visual SLAM at large scales. In *IEEE IROS*. [3.2.3, 7.2]

[Meilland et al., 2011] Meilland, M., Comport, A., and Rives, P. (2011). Dense visual mapping of large scale environments for real-time localisation. In *IEEE IROS*. [2.2]

[Meilland et al., 2015] Meilland, M., Comport, A., and Rives, P. (2015). Dense omnidirectional RGB-D mapping of large-scale outdoor environments for real-time localization and autonomous navigation. *JFR*, 32(4). [2.1, 2.3.3, 4.2.2.1, 5.1, 5.4.3, 7.1, 7.2, 7.4.1]

[Meilland et al., 2013] Meilland, M., Drummond, T., and Comport, A. (2013). A unified rolling shutter and motion blur model for 3D visual registration. In *IEEE ICCV*. [1]

[Melekhov et al., 2017] Melekhov, I., Ylioinas, J., Kannala, J., and Rahtu, E. (2017). Relative camera pose estimation using convolutional neural networks. [8.1, 1]

[Merrell et al., 2007] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J., Yang, R., Nister, D., and Pollefeys, M. (2007). Real-time visibility-based fusion of depth maps. In *IEEE ICCV*. [7.3.3]

[Micušık et al., 2003] Micušık, B., Martinec, D., and Pajdla, T. (2003). 3D metric reconstruction from uncalibrated omnidirectional images. In *ACCV*. [2.1]

[Mitra and Nguyen, 2003] Mitra, N. and Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *ACM SCG*. [2.5]

[Morency and Darrell, 2002] Morency, L. and Darrell, T. (2002). Stereo tracking using ICP and normal flow constraint. In *ICPR*. [3.2, 3.3.2, 5.1.1, 5.1.2, 5.2.1]

[Muller et al., 2011] Muller, T., Rannacher, J., Rabe, C., and Franke, U. (2011). Feature- and depth-supported modified total variation optical flow for 3D motion field estimation in real scenes. In *IEEE CVPR*. [3.4, 5.1.1, 6.3.1]

[Munoz and Comport, 2016a] Munoz, F. I. and Comport, A. (2016a). Point-to-hyperplane RGB-D pose estimation: Fusing photometric and geometric measurements. In *IEEE IROS*. [3.3.2, 5.1.1]

[Munoz and Comport, 2016b] Munoz, F. I. and Comport, A. (2016b). A proof that fusing measurements using point-to-hyperplane registration is invariant to relative scale. In *IEEE MFI*. [5.1.1]

[Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J., and Tardós, J. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5). [3.1, 3.2.1, 3.2, 7.2]

[Newcombe et al., 2011a] Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011a). KinectFusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*. [3.2.3, 3.2.3, 5.1, 5.4, 7.2]

[Newcombe et al., 2011b] Newcombe, R., Lovegrove, S., and Davison, A. (2011b). DTAM: dense tracking and mapping in real-time. In *IEEE ICCV*. [6.2, 7.2]

[Nicolai et al., 2016] Nicolai, A., Skeele, R., Eriksen, C., and Hollinger, G. (2016). Deep learning for laser-based odometry estimation. In *RSS Workshop on Limits and Potentials of Deep Learning in Robotics*. [8.1, 1]

[Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *IEEE CVPR*. [3.2, 3.2.1]

[Ogniewicz, 1994] Ogniewicz, R. (1994). Skeleton-space: a multiscale shape description combining region and boundary information. In *IEEE CVPR*. [7.3.2]

[Ogniewicz and Ilg, 1992] Ogniewicz, R. and Ilg, M. (1992). Voronoi skeletons: theory and applications. In *IEEE CVPR*. [7.3.2]

[Pérez-Yus et al., 2016] Pérez-Yus, A., López-Nicolás, G., and Guerrero, J. (2016). Peripheral expansion of depth information via layout estimation with fisheye camera. In *IEEE ECCV*. [2.1]

[Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. PAMI*, 12(7). [6.2]

[Pinies et al., 2015] Pinies, P., Paz, L., and Newman, P. (2015). Too much TV is bad: Dense reconstruction from sparse laser with non-convex regularisation. In *IEEE ICRA*. [6.3.1]

[Pomerleau et al., 2015] Pomerleau, F., Colas, F., and Siegwart, R. (2015). A review of point cloud registration algorithms for mobile robotics. *Found. Trends Robot*, 4(1). [3.2.3, 3.3.2, 4.1]

[Puig et al., 2012] Puig, L., Bermúdez, J., Sturm, P., and Guerrero, J. (2012). Calibration of omnidirectional cameras in practice: A comparison of methods. *CVIU*, 116(1). [2.2, 5.2]

[Quiroga et al., 2014] Quiroga, J., Brox, T., Devernay, F., and Crowley, J. (2014). Dense semi-rigid scene flow estimation from RGBD images. In *IEEE ECCV*. [3.3, 3.2.3, 3]

[Rosten et al., 2010] Rosten, E., Porter, R., and Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE Trans. on PAMI*, 32(1). [3.2.1]

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *IEEE ICCV*. [3.2.1]

[Rusu et al., 2007] Rusu, R., Blodow, N., Marton, Z., Soos, A., and Beetz, M. (2007). Towards 3D object maps for autonomous household robots. In *IEEE IROS*. [2.5]

[Sacht et al., 2010] Sacht, L., Carvalho, P., Velho, L., and Gattass, M. (2010). Face and straight line detection in equirectangular images. In *Workshop de Visao Computacional*. [2.2.2.1]

[Salaris et al., 2017] Salaris, P., Spica, R., Robuffo Giordano, P., and Rives, P. (2017). Online Optimal Active Sensing Control. In *IEEE ICRA*. [8.1, 1]

[Saurer et al., 2015] Saurer, O., Pollefeys, M., and Lee, G. (2015). A minimal solution to the rolling shutter pose estimation problem. In *IEEE IROS*. [1]

[Scandaroli et al., 2012] Scandaroli, G., Meilland, M., and Richa, R. (2012). Improving NCC-based direct visual tracking. In *IEEE ECCV*. [3.2.2]

[Scaramuzza and Fraundorfer, 2011] Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry [tutorial]. *IEEE Robot. Automat. Mag.*, 18(4). [3.1]

[Scaramuzza and Siegwart, 2008] Scaramuzza, D. and Siegwart, R. (2008). Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Trans. on Robotics*, 24(5). [2.1, 4.1.1, 4.2.1.2]

[Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3). [3.1]

[Schönbein and Geiger, 2014] Schönbein, M. and Geiger, A. (2014). Omnidirectional 3D reconstruction in augmented Manhattan worlds. In *IEEE IROS*. [2.3.2, 6.2]

[Serafin and Grisetti, 2015] Serafin, J. and Grisetti, G. (2015). NICP: dense normal based point cloud registration. In *IEEE IROS*. [4.1]

[Shamir, 2008] Shamir, A. (2008). A survey on mesh segmentation techniques. *Computer Graphics*. [6.2]

[Siddiqi and Pizer, 2008] Siddiqi, K. and Pizer, S. (2008). *Medial Representations: Mathematics, Algorithms and Applications*. Springer Publishing Company, Incorporated. [7.3.2]

[Silveira, 2014] Silveira, G. (2014). Photogeometric direct visual tracking for central omnidirectional cameras. *J. Math. Imaging Vis.*, 48(1). [3.2, 3.2.2, 3.3.2, 5.1]

[Silveira et al., 2008] Silveira, G., Malis, E., and Rives, P. (2008). An efficient direct approach to visual SLAM. *IEEE Trans. on Rob.*, 24(5). [7.2]

[Singh et al., 2017] Singh, A., Siam, M., and Jägersand, M. (2017). Unifying registration based tracking: A case study with structural similarity. In *IEEE WACV*. [3.2.2]

[Snavely et al., 2006] Snavely, N., Seitz, S., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph.*, 25(3). [3.1, 7.2]

[Stoyanov et al., 2012] Stoyanov, T., Magnusson, M., Andreasson, H., and Lilienthal, A. (2012). Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *IJRR*, 31(12). [4.1.1]

[Strasdat et al., 2010] Strasdat, H., Montiel, J., and Davison, A. (2010). Scale drift-aware large scale monocular SLAM. In *RSS*. [2]

[Timofte and Gool, 2015] Timofte, R. and Gool, L. V. (2015). Sparse flow: Sparse matching for small to large displacement optical flow. In *IEEE WCACV*. [5.1.1]

[Torii et al., 2009] Torii, A., Havlena, M., and Pajdla, T. (2009). From Google Street view to 3D city models. In *IEEE ICCV Workshops*. [2.1]

[Tykkala et al., 2011] Tykkala, T., Audras, C., and Comport, A. (2011). Direct iterative closest point for real-time visual odometry. In *ICCV Workshops*. [3.2, 3.2.3, 3.3.2, 5.1, 5.1.1, 5.1.2, 5.2.1, 5.5, **??**, 5.4, 5.1, 5.2, 5.10, 5.3, 5.4, 6.5.1, 6.11]

[Ulrich and Nourbakhsh, 2000] Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *IEEE ICRA*, volume 2. [2.1]

[Vasconcelos et al., 2012] Vasconcelos, F., Barreto, J., and Nunes, U. (2012). A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE Trans. on PAMI*, 34(11). [2.3.1]

[Vázquez-Otero et al., 2015] Vázquez-Otero, A., Faigl, J., Dormido, R., and Duro, N. (2015). Reaction diffusion voronoi diagrams: From sensors data to computing. *Sensors*, 15(6). [7.3.2]

[Vedaldi and Fulkerson, 2008] Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/. [3.2.1]

[Vogel et al., 2013] Vogel, C., Schindler, K., and Roth, S. (2013). Piecewise rigid scene flow. In *IEEE ICCV*. [6.2]

[Wang et al., 2014] Wang, Q., Yu, Z., Rasmussen, C., and Yu, J. (2014). Stereo vision based depth of field rendering on a mobile device. In *JEI*. [6.3.1]

[Wang et al., 2016] Wang, W., Sakurada, K., and Kawaguchi, N. (2016). Incremental and enhanced scanline-based segmentation method for surface reconstruction of sparse LiDAR data. *Remote Sensing*, 8(11). [6.2, 7.2]

[Wasenmüller and Stricker, 2016] Wasenmüller, O. and Stricker, D. (2016). Comparison of kinect V1 and V2 depth images in terms of accuracy and precision. In *ACCV*. [3.2.3]

[Weikersdorfer et al., 2013] Weikersdorfer, D., Gossow, D., and Beetz, M. (2013). Depth-adaptative superpixels. In *IEEE ICP*. [6.2]

[Weinzaepfel et al., 2013] Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. In *IEEE ICCV*. [3.2.3]

[Weyand et al., 2016] Weyand, T., Kostrikov, I., and Philbin, J. (2016). PlaNet - photo geolocation with convolutional neural networks. In *IEEE ECCV*. [8.1, 1]

[Whelan et al., 2015] Whelan, T., Kaess, M., Johannsson, H., Fallon, M., Leonard, J., and McDonald, J. (2015). Real-time large scale dense RGB-D SLAM with volumetric fusion. *IJRR*, 34. [3.1, 7.1, 7.2]

[Wu, 2013] Wu, C. (2013). Towards linear-time incremental structure from motion. In *3DV*. [3.1, 1, 7.2]

[Yamaguchi et al., 2014] Yamaguchi, K., McAllester, D., and Urtasun, R. (2014). Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *IEEE ECCV*. [2.3.2, 6.2]

[Yang et al., 2016] Yang, J., Li, H., Campbell, D., and Jia, Y. (2016). Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. on PAMI*, 38(11). [3.3.3.2, 4.1.1, 4.5]

[Zelnik-Manor et al., 2005] Zelnik-Manor, L., Peters, G., and Perona, P. (2005). Squaring the circles in panoramas. In *IEEE ICCV*. [2.2.2.1]

[Zhang and Pless, 2004] Zhang, Q. and Pless, R. (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE IROS*. [2.3.1]

[Zhang et al., 2012] Zhang, Q., Ye, M., Yang, R., Matsushita, Y., and Wilburn, B. (2012). Edge-preserving photometric stereo via depth fusion. In *IEEE CVPR*. [6.2]

[Zhang, 1995] Zhang, Z. (1995). Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report 2676, Inria. [3.3.3, 4.2.2, 5.2, 5.3.2.1, A.4]

[Zhang, 1997] Zhang, Z. (1997). Motion and structure from two perspective views: from essential parameters to euclidean motion through the fundamental matrix. *J. Opt. Soc. Am. A*, 14(11). [2, 3.2.2]

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. PAMI*, 22(11). [2.2.1, 10]

[Zhang et al., 2016] Zhang, Z., Rebecq, H., Forster, C., and Scaramuzza, D. (2016). Benefit of large field-of-view cameras for visual odometry. In *IEEE ICRA*. [2.1, 3.2, 3.3.2, 4.4.2, 6.4.2.1, 6.4.2.1]

[Zhou et al., 2016a] Zhou, Y., Kneip, L., and Li, H. (2016a). Real time rotation estimation for dense depth sensors in piece-wise planar environments. In *IEEE IROS*. [4.1, 4.1.1, 4.1.2, 4.2.1.4]

[Zhou et al., 2016b] Zhou, Y., Kneip, L., Rodriguez, C., and Li, H. (2016b). Divide and conquer: Effcient density-based tracking of 3D sensors in Manhattan worlds. In *ACCV*. [4.1.1]

Résumé

Cette thèse se situe dans le domaine de l'auto-localisation et de la cartographie 3D pour des robots mobiles et des systèmes autonomes avec des caméras RGB-D. Nous présentons des techniques d'alignement d'images et de cartographie pour effectuer la localisation d'une caméra (suivi), notamment pour des caméras avec mouvements rapides ou avec faible cadence. Les domaines d'application possibles sont la réalité virtuelle et augmentée, la localisation de véhicules autonomes ou la reconstruction 3D des environnements.

Nous proposons un cadre consistant et complet au problème de localisation et cartographie 3D à partir de séquences d'images RGB-D acquises par une plateforme mobile. Ce travail explore et étend le domaine d'applicabilité des approches de suivi direct dites "appearance-based". Vis-à-vis des méthodes fondées sur l'extraction de primitives, les approches directes permettent une représentation dense et plus précise de la scène mais souffrent d'un domaine de convergence plus faible nécessitant une hypothèse de petits déplacements entre images.

Dans la première partie de la thèse, deux contributions sont proposées pour augmenter ce domaine de convergence. Tout d'abord une méthode d'estimation des grands déplacements est développée s'appuyant sur les propriétés géométriques des cartes de profondeurs contenues dans l'image RGB-D. Cette estimation grossière (rough estimation) peut être utilisée pour initialiser la fonction de coût minimisée dans l'approche directe. Une seconde contribution porte sur l'étude des domaines de convergence de la partie photométrique et de la partie géométrique de cette fonction de coût. Il en résulte une nouvelle fonction de coût exploitant de manière adaptative l'erreur photométrique et géométrique en se fondant sur leurs propriétés de convergence respectives.

Dans la deuxième partie de la thèse, nous proposons des techniques de régularisation et de fusion pour créer des représentations précises et compactes de grands environnements. La régularisation s'appuie sur une segmentation de l'image sphérique RGB-D en patchs utilisant simultanément les informations géométriques et photométriques afin d'améliorer la précision et la stabilité de la représentation 3D de la scène. Cette segmentation est également adaptée pour la résolution non uniforme des images panoramiques. Enfin les images régularisées sont fusionnées pour créer une représentation compacte de la scène, composée de panoramas RGB-D sphériques distribués de façon optimale dans l'environnement. Ces représentations sont particulièrement adaptées aux applications de mobilité, tâches de navigation autonome et de guidage, car elles permettent un accès en temps constant avec une faible occupation de mémoire qui ne dépendent pas de la taille de l'environnement.

Mots Clés

Recalage d'images; cartographie; odométrie visuelle; localisation; SLAM visuel; images panoramiques

Abstract

This thesis is in the context of self-localization and 3D mapping from RGB-D cameras for mobile robots and autonomous systems. We present image alignment and mapping techniques to perform the camera localization (tracking) notably for large camera motions or low frame rate. Possible domains of application are localization of autonomous vehicles, 3D reconstruction of environments, security or in virtual and augmented reality.

We propose a consistent localization and 3D dense mapping framework considering as input a sequence of RGB-D images acquired from a mobile platform. The core of this framework explores and extends the domain of applicability of direct/dense appearance-based image registration methods. With regard to feature-based techniques, direct/dense image registration (or image alignment) techniques are more accurate and allow us a more consistent dense representation of the scene. However, these techniques have a smaller domain of convergence and rely on the assumption that the camera motion is small.

In the first part of the thesis, we propose two formulations to relax this assumption. Firstly, we describe a fast pose estimation strategy to compute a rough estimate of large motions, based on the normal vectors of the scene surfaces and on the geometric properties between the RGB-D images. This rough estimation can be used as initialization to direct registration methods for refinement. Secondly, we propose a direct RGB-D camera tracking method that exploits adaptively the photometric and geometric error properties to improve the convergence of the image alignment.

In the second part of the thesis, we propose techniques of regularization and fusion to create compact and accurate representations of large scale environments. The regularization is performed from a segmentation of spherical frames in piecewise patches using simultaneously the photometric and geometric information to improve the accuracy and the consistency of the scene 3D reconstruction. This segmentation is also adapted to tackle the non-uniform resolution of panoramic images. Finally, the regularized frames are combined to build a compact keyframe-based map composed of spherical RGB-D panoramas optimally distributed in the environment. These representations are helpful for autonomous navigation and guiding tasks as they allow us an access in constant time with a limited storage which does not depend on the size of the environment.

Keywords

RGB-D registration; mapping; visual odometry; localization; visual SLAM; panoramic images