DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Xi WANG

Improving robustness of monocular visual SLAM techniques

Influence of contrasted local features, multi-planar representations and multi-modal image analysis

Thèse présentée et soutenue à Inria Rennes/IRISA, le TBA
Unité de recherche : IRISA
Thèse N° : TBA

Rapporteurs avant soutenance :

Cédric DEMONCEAUX	Professeur à l'Université de Bourgogne
José María Martínez Montiel	Professeur à l'Université de Saragosse

Composition du Jury :

Président(e) :		
Examinateurs :	Andrew COMPORT	Chargé de Recherche - Centre Nationale de Recherche Scientifique
	Cédric DEMONCEAUX	Professeur à l'Université de Bourgogne
	Luce MORIN	Professeure des Universités à l'INSA Rennes
	Marc CHRISTIE	Maitre de Conférence à l'Université de Rennes 1
	José María Martínez Montiel	Professeur à l'Université de Saragosse
	Timothée DE GOUSSENCOURT	CTO à l'entreprise oARo-Solidanim
Dir. de thèse :	Éric MARCHAND	Professeur des Universités à l'Université de Rennes 1

Acknowledgement

I'd like to first appreciate all the member in the committee for accepting to be in the jury and taking time and efforts on reviewing my thesis, and all the valuable and concrete advices during the defence. I d like specially present my gratitude to Prof Jose Maria Martinez Montiel and Prof Cedric Demonceaux for review my thesis and writing the report.

Then I'd take this opportunity to speak by myself and present my gratitude to both of my supervisors as all I have done would not have been even possible without the support of them, they guided me from the confusion, provided numerous valuable insights and feedbacks across all the discussions, basically they taught me how to do research from step to step.

I'd like to thank Eric Marchand for his valuable inputs and feedbacks, in-depth understanding and expertise of the problem in the domain of robotics and computer vision. Most importantly his rigorous working attitude really inspired me during all my Ph.D. time.

And I also like to thank Marc Christie for his sharp insight, rich knowledge on camera control and animation, excellent skills on writing and presenting for all the publications and even the thesis, and also his sense of humour. He is truly one of the coolest researchers I have ever met.

Also big thanks to Tim De Goussencourt for his engineering experience and all the techniques and knowledges in the filmmaking and broadcasting world, really opens a new door for me. I appreciate all his support during the time when we were in Bordeaux and all the discussions and progress on the product and algorithms.

I'd also like to thanks all my colleagues for the discussion, ideas and the coffees.

Finally I wish to present all of my gratitudes to all my colleagues, my friends and my parents, who supported me all the time from China to France, from days to nights. As there are too many thanks to say and too little space to write.

I'd never expect my experience of Ph.D. to be like this one, I am so grateful that I could spend some time of my life with everyone I met and everything I've done.

Table of Contents

Chapter 1

Introduction

A Robot Needs eyes

Man is the measure of all things: of the things that are, that they are, of the things that are not, that they are not said by Protagoras of Abdera, was regarded as one of his most famous statements for depicting the relativism, anthropocentrism and even inspiring the idealism. Interestingly, one can borrow this famous yet old line then rejuvenate it in totally different circumstances: robot agents. If we decide to reinterpret the sentence with a modern mathematical and scientific view, a regard towards the robotics research domain can be taken place from an anthropology vantage: like a human being or other animals, before all the possible actions, a robot needs to recognize, measure and understand the around environment, such that correct decisions can be made and intelligent actions can be triggered. The measurement of the world is not realized by a pure concept or direct interaction, but through various entitative devices or organs as mediums for acquiring and transporting information from the real natural world into the symbolic and conceptual layer, where the knowledge and intelligence form their existence.

When confronting the complexe natural environment, human being measures it with their most efficient and fast visual organs: eyes. Our visual system provides countless information and has already become one of the most critical perception devices to support our daily life. From scenarios of recognizing the neighbours and saying hello in the morning, to the missions of navigating oneself on a jammed and even sometimes dangerous highway to home, our visual organs faithfully accomplish their duty and assist us with active visual information for almost every single decision we make.

As human beings created robot agents in our own image, a similar thinking paradigm can undoubtedly be found in these intelligent machines. From science fiction <Robot Visions> of Issac Asimov to Hollywood films <Articificial Intelligence> by Steven Spielberg, human society places robot agents in the future perspectives like an alter ego of ourselves and wishes that they could eventually help alleviate human workloads. In all the dreams,

it requires our smart android friends to act and move, most importantly, understand and think. To build the bridge of understanding from the exterior environment to the interior system, one can even assert with confidence that the basic perception devices and functionalities of capturing the surrounding environment should be regarded as one of the most fundamental and mandatory modules on a robot agent. In other words, a robot needs eyes.

History and Development of Robotics Visual Systems

Actually, the definition of robot has been proposed for quite a long time. The idea of automata originates in the mythologies of many cultures throughout the history. From ancient China, Mozi and Lu-Ban created self-operation birds and humanoids, to the speaking machine built by Hero of Alexandria, the very concept of robot agents rooted in all civilizations, and haunted numerous great engineers and scientists in their dreams or nightmares. However, almost all the famous robot inventions in history did not equip perceptional devices for the abstruse science dependencies and sophisticated technologies ahead of time. The last piece of the puzzle was finally completed with the rapid development of semiconductor and information technologies across the second and the third industrial revolutions. Human being created our own artificial visual system in the 20th century, the digital camera. A digital camera system composes image sensors crafted by the array of photon transistors and auxiliary electronic systems, which could finally make machines capture the illuminative images in digital format and enable robots to see the world to eventually understand it.

To achieve this goal, multiple new domains and communities were spawned and gained worldwide popularity in scientific research and industrial companies. These communities include image processing, machine vision, robotics, computer vision, deep learning and computer graphics, etc., among which computer vision and robotics directly contribute to the robotics visual system and make the robot observe the illuminative world. The main idea can be abstractly introduced as using digital camera devices and algorithms to reconstruct and track a robot agent in an unknown environment. Some people utilize the *Kidnapped Robot Problem* to explain the situation where an autonomous robot is placed into an arbitrary location and requires to localize itself and understand the surrounding environment with mounted sensors (*eg.* digital cameras).

Through the development of modern robotics technologies, multiple methods, sensors and even combinations of sensors are proposed to rescue the *kidnapped* robot agents, which

(a) Apple AR (Augmented Reality) (b) Autonomous vehicles

Figure 1.1 – (a) Demonstration of augmented reality application on Apple iPad device for showing decorations of layout. (b) In autonomous driving vehicle systems, SLAM, motion planning, detection algorithms are kernel functionalities.

we will elaborate in the following chapters of background and related works. Beyond this hypothetical scenario of saving robots from unknown circumstances, more practical applications and techniques benefitted from robotics and computer vision progress. One classic example is the main topic of this thesis: the SLAM (Simultaneous Localization And Mapping) technique. SLAM technique concentrates on localizing and recovering the environment in a simultaneous way and is one of the core functionalities of many industrial products such as augmented reality (Fig. 1.1a), where the device poses should be tracked in real-time; autonomous driving (Fig. 1.1b), where one needs to localize the vehicle in a pre-generated map or unknown environment; and even modern filmmaking workflow, where the relative camera position and orientation are critical for post-processing or real-time prevising for directors and actors to visualise the visual effects on the stage.

Why is the Robotics Vision Hard?

Multiple difficulties in different levels can influence the final performance of robot agents's SLAM task, as the pipeline is long and complicated from the real world physics to the required information such as agent poses and 3-D map, which help us visualize colourful graphics scenes in AR devices or make hard decisions on the highway for autonomous driving.

As the digital camera acquires information from the physical world and reinterpret them into digital format, *i.e.* pixels, many compromises have been made to make sure the whole workflow is feasible.

For articulating this problem correctly, we categorize critical factors into two genres

and give more detailed discussions respectively: intrinsic limits and extrinsic noises.

For the intrinsic limits, it refers to the unknown or simplified factors when doing the mathematical modelling. For example, an image is actually influenced by multiple factors, not only the geometry factor but also the illuminative factor and material properties, even the kinematic conditions of the scene: it makes the objects look drastically dissimilar under different lighting conditions, also highly dependent on the material itself, and much more complicated if we are not sure the object's motion condition *i.e.* dynamic objects. For example, a metal material sphere and a wooden material one demonstrate different illuminative information under sunlight though their geometric structures are very similar. Another examples can be given for the famous day-night shift difference, in which the changes are not only on the brightness in total but also the relative shading and chromatic information between each pixel region (Fig. 1.2). When applying the inference towards our interested information without knowing the scene, the kinematic conditions, and material properties (which is difficult to know from a priori), numerous ambiguities and ill-posed conditions may occur and pollute the robot agent's computation.

The extrinsic noises usually mean the error or imprecision when measuring and observing the exterior world, often caused by wrong judgment or engineering limits and randomly generated noises in the electronic and optical deivces. Errors may come from the current electronic equipment and physical constraints, such as the rounding error due to the limited size of the physical pixel on the sensor; random pixel noises when augmenting the ISO value in the dark condition; the distortion and aberrations caused by optic flaws; and the compromised limited binary digits for economizing device storage, etc.

Many solutions are proposed for addressing each problem, respectively, with the means from classic statistic probability models to the modern data-driven deep neural network. However, the quest of improving the robot's robustness under dynamic and complicated environments perisists and becomes more and more significant and active for nowadays robotics reseach. The need for improving the robustness of robot agents is imminent and regarded as one of most imperative factors for deploying robots ubiquitously in our daily life.

What does this Thesis Discuss?

Under this context, this thesis tries to address a small drop in the ocean of the problem of SLAM robustness, yet in a very systematic view: we try to break down the SLAM system into different and inter-influential modules. Then use the concept of "divide and

(a) Summerday (b) Snow (c) Night Rainy

Figure 1.2 – The influence of seasonal and lighting condition on acquired digital images, the pictures are taken under the same location but during summer, snowing and night rainy days respectively. Images from Oxford Robotcar Dataset [70]

conquer" for answering possible questions within each module and wishing to contribute to the community and help rescue as soon as possible the *kidnapped robot*.

With the above objectives, the contributions of the thesis are stated as follows for tackling the robustness problem from multiple angles:

1. From the image feature angle, we proposed a multiple layered image structure for improving the performance of traditional local image features under extreme conditions. Furthermore, an optimization method on linear searching and mutual information assisted convex optimization are designed for tuning the optimal parameters with the proposed structure.
2. From the geometric primitive angle, we proposed a relative pose estimation and SLAM framework under the multiple planar assumption, by keypoint feature-based and template tracker based methods, respectively. We tried to achieve better performance of mapping and tracking simultaneously with the help of a more general planar assumption.
3. From the angle of relocalization of the SLAM system, the idea is to recover the already passed locations of the robot agent for lowering the overall estimation error or when the robot is in lost status. We proposed a binary graph structure for embedding spatial information and heterogeneous data formats such as depth image, semantic information etc. The proposed method enables robotics SLAM systems to relocalize themselves with a higher success rate even under different lighting, weather and seasonal conditions.

Thesis Relative Publications

1. Xi Wang, Marc Christie, and Eric Marchand, « Multiple Layers of Contrasted Images for Robust Feature-Based Visual Tracking », *in*: *IEEE Int. Conf. on Image Processing (ICIP)*, 2018
2. Xi Wang, Marc Christie, and Eric Marchand, « Optimized Contrast Enhancements to Improve Robustness of Visual Tracking in a SLAM Relocalisation Context », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018
3. Xi Wang, Marc Christie, and Eric Marchand, « Relative Pose Estimation and Planar Reconstruction via Superpixel-Driven Multiple Homographies », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020
4. Xi Wang, Marc Christie, and Eric Marchand, « TT-SLAM: Dense Monocular SLAM for Planar Environments », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Xi'an, China, 2021
5. Xi Wang, Marc Christie, and Eric Marchand, « Binary Graph Descriptor for RobustRelocalization on Heterogeneous Data », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2022)

Chapter 2

Résumé

Un Robot a Besoin d'Yeux

L'homme est la mesure de toutes choses : de celles qui sont, du fait qu'elles sont ; de celles qui ne sont pas, du fait qu'elles ne sont pas. dit par Protagoras de Abdera, était considéré comme l'un de ses arguments les plus célèbres pour dépeindre le relativisme, l'anthropocentrisme et même inspirer l'idéalisme. De façon intéressante, on peut emprunter ce fameux et pourtant ancienne fragment puis la rajeunir dans des circonstances totalement différentes : les agents robotique. Si nous décidons de réinterpréter la phrase avec une vision mathématique et scientifique moderne, un regard vers le domaine de recherche en robotique peut être pris du point de vue de l'anthropologie : comme un être humain ou d'autres animaux, avant toutes les actions possibles, un robot doit reconnaître, mesurer et comprendre l'environnement extérieur, de sorte que les bonnes décisions puissent être prises et que des actions intelligentes puissent être déclenchées. La mesure du monde n'est pas réalisée par un concept pur ou une interaction directe, mais à travers divers dispositifs ou organes entitatifs en tant que millieu pour acquérir et transporter des informations du monde naturel réel dans le domaine symbolique et conceptuelle, où la connaissance et l'intelligence forment leur existence.

Face à l'environnement naturel riche et immersif, l'être humain mesure une grande partie des choses avec ses organes visuels les plus efficaces et les plus rapides: les yeux. Notre système visuel fournit d'innombrable information et il est déjà devenu l'un des dispositifs de perception les plus critiques pour soutenir notre vie quotidienne. Des scénarios de reconnaissance des voisins au matin aux missions de navigation sur une autoroute pour rentrer chez soi, nos organes visuels accomplissent fidèlement leur devoir et nous assistent avec des informations visuelles riches et actives pour presque tous les décision que nous prenons.

Comme les êtres humains ont créé des agents robotiques à notre image, un paradigme de pensée similaire peut être sans doute trouvé dans ces machines intelligentes. De la

science-fiction <Robot Visions> d'Issac Asimov aux films Hollywood <Articificial Intelligence> de Steven Spielberg, la société humaine place les agents robotiques dans l'imagination future comme un alter ego de nous-mêmes et souhaite qu'ils puissent éventuellement aider à alléger les charges de travail humaines. Dans tous les rêves, il faut que nos amis androïdes intelligents agissent et bougent, surtout, comprennent et réfléchissent. Pour construire le pont de compréhension de l'environnement extérieur au système intérieur, on peut même affirmer avec certitude que les dispositifs de perception de base et les fonctionnalités de capture de l'environnement entour doivent être considérés comme l'un des modules les plus fondamentaux et nécessaires d'un agent robotique. En d'autres termes, un robot a besoin d'yeux.

Histoire et Développement des Systèmes Visuels de Robotique

En fait, la définition de robot est proposée depuis longtemps. L'idée d'automates trouve son origine dans les mythologies de nombreuses cultures à travers l'histoire. De la Chine ancienne, Mozi et Lu ban ont créé des oiseaux et des humanoïdes autonomes, jusqu'à la machine parlante construite par Hero de Alexandria, le concept de robots a enracinés dans toutes les civilisations et a hanté de nombreux grands ingénieurs et scientifiques dans leurs rêves ou cauchemars. Cependant, presque toutes les inventions de robots célèbres dans l'histoire n'ont pas équipé des dispositifs de perception pour les dépendances scientifiques absconses et les technologies sophistiquées. La dernière pièce du puzzle a finalement été complétée avec le développement rapide des technologies des semi-conducteurs et de l'information au cours de la deuxième et de la troisième révolution industrielle. L'être humain a créé notre système visuel artificiel au 20e siècle, l'appareil photo numérique. Un système d'appareil photo numérique compose des capteurs d'images fabriqués par la matrice de transistors à photons et de systèmes électroniques auxiliaires, qui pourraient enfin permettre aux machines de capturer les images illuminatives au format numérique et rendre les robots possible de voir le monde pour éventuellement le comprendre.

Pour atteindre cet objectif, plusieurs nouveaux domaines et communautés ont été créés et ont gagné en popularité dans le monde entier de la recherche scientifique aux entreprises industrielles. Ces communautés comprennent le traitement d'images, la robotique, la vision par ordinateur, le deep learning et l'infographie, etc., parmi lesquelles la vision par ordinateur et la robotique contribuent directement au système visuel de la robotique et font observer au robot le monde illuminatif. L'idée principale peut être présentée de

manière abstraite comme l'utilisation d'appareils photo numériques et d'algorithmes pour reconstruire et suivre un agent robot dans un environnement inconnu. Certaines personnes utilisent la description: *le Problème du Robot Kidnappé* pour expliquer la situation où un robot autonome est placé dans un endroit arbitraire et doit se localiser et comprendre l'environnement autour avec des capteurs montés (*eg.* des caméras numériques).

Grâce au développement des technologies robotiques modernes, de multiples méthodes, capteurs et même combinaisons de capteurs sont proposés pour sauver les agents robots *kidnappé*, que nous développerons dans les chapitres suivants sur le contexte. Au-delà de ce scénario hypothétique consistant à sauver les robots de circonstances inconnues, des applications et des techniques plus pratiques ont bénéficié des progrès de la robotique et de la vision par ordinateur. Un exemple classique est le sujet principal de cet article : la technique SLAM (Simultaneous Localization And Mapping). La technique SLAM se concentre sur la localisation et la récupération de l'environnement de manière simultanée et est l'une des fonctionnalités de base de nombreux produits industriels tels que la réalité augmentée, où les poses de l'appareil doivent être suivies dans temps réel; conduite autonome, où il faut localiser le véhicule dans une carte pré-générée ou un environnement inconnu ; et même le flux de travail cinématographique moderne, où la position et l'orientation relatives de la caméra sont essentielles pour le post-traitement ou la prévision en temps réel permettant aux réalisateurs et aux acteurs de visualiser les effets visuels sur la scène.

Pourquoi la Vision Robotique est-elle Difficile ?

De multiples difficultés dans les différentes layers peuvent influencer la performance finale de la tâche SLAM des agents robotiques, car le pipeline est long et compliqué de la physique du monde réel aux informations requises telles que les poses des agents et la carte 3D, qui nous aident à visualiser des scènes graphiques colorées dans les appareils RA (Réalité Augmenté) ou prenez des décisions difficiles sur l'autoroute pour une véhicule autonome.

Au fur et à mesure que l'appareil photo numérique acquiert les informations du monde physique et les réinterprète au format numérique, *i.e.* en pixels, de nombreux compromis ont été faits pour s'assurer que l'ensemble du flux de travail est réalisable.

Pour articuler correctement ce problème, nous catégorisons les facteurs critiques en deux genres et donnons respectivement des discussions plus détaillées : les limites intrinsèques et les bruits extrinsèques.

Pour les limites intrinsèques, il fait référence aux facteurs inconnus ou simplifiés lors de la modélisation mathématique. Par exemple, une image est en fait influencée par de multiples facteurs, non seulement le facteur géométrique, mais aussi le facteur d'éclairage et les propriétés du matériau, voire les conditions cinématiques de la scène : cela rend les objets radicalement différents dans différentes conditions d'éclairage, également fortement dépendant de le matériau per sé, et bien plus compliqué si l'on n'est pas sûr de la condition de mouvement de l'objet *eg.* des objets dynamiques. Par exemple, une sphère en métal et une en bois présentent des informations lumineuses différentes sous la lumière du soleil malgré leurs structures géométriques soient très similaires. Un autre exemple peut être donné pour la fameuse différence de transition jour-nuit, dans laquelle les changements ne concernent pas seulement la luminosité totale mais aussi l'ombrage relatif et les informations chromatiques entre chaque région de pixels. Lorsque l'on applique l'inférence à nos informations intéressées sans connaître la scène, les conditions cinématiques et les propriétés des matériaux (ce qui est difficile à connaître à priori), de nombreuses ambiguïtés et conditions mal-posées peuvent survenir et polluer le calcul de l'agent robot.

Les bruits extrinsèques signifient généralement l'erreur ou l'imprécision lors de la mesure et de l'observation du monde extérieur, souvent causées par un mauvais jugement ou des limites techniques et des bruits générés de manière aléatoire dans les appareils électroniques et optiques. Les erreurs peuvent provenir de l'équipement électronique actuel et des contraintes physiques, telles que l'erreur d'arrondi due à la taille limitée du pixel physique sur le capteur ; bruits de pixels aléatoires lors de l'augmentation de la valeur ISO dans l'obscurité ; la distorsion et les aberrations causées par les défauts optiques ; et les chiffres binaires limités compromis pour économiser le stockage de l'appareil, etc.

De nombreuses solutions sont proposées pour résoudre chaque problème, respectivement, avec les moyens des modèles de probabilité statistiques classiques au moderne deep learning basé sur les données. Cependant, la quête d'amélioration de la robustesse du robot dans des environnements dynamiques et complexes persiste et devient de plus en plus importante et active pour la recherche en robotique d'aujourd'hui. Le besoin d'améliorer la robustesse des agents robots est immnent et considéré comme l'un des facteurs les plus impératifs pour déployer des robots de manière omniprésente dans notre vie quotidienne.

Qu'est-ce que cette thèse discute ?

Dans ce contexte, cette thèse tente d'aborder une petite goutte dans l'océan du problème de la robustesse du SLAM, mais dans une vision très systématique : nous essayons de décomposer le système SLAM en modules différents et inter-influents. Utilisez ensuite le concept de « diviser pour mieux régner » pour répondre aux questions au sein de chaque module et souhaiter contribuer à la communauté et aider à sauver le plus rapidement possible notre *robot kidnappé*.

Avec les objectifs ci-dessus, les contributions de la thèse sont énoncées comme suit pour aborder le problème de robustesse sous plusieurs angles :

1. Du point de vue de l'image, nous avons proposé une structure d'image à plusieurs layers pour améliorer les performances des caractéristiques d'image locales traditionnelles dans des conditions extrêmes. De plus, une méthode d'optimisation sur la recherche linéaire et l'optimisation convexe assistée par information mutuelle sont conçues pour régler les paramètres optimaux avec la structure proposée.
2. Du point de vue du primitif géométrique, nous avons proposé une estimation de pose relative et un cadre SLAM sous l'hypothèse de plans multiples, respectivement par des méthodes basées sur des caractéristiques de points clés et basées sur des modèles de suivi. Nous avons essayé d'obtenir de meilleures performances de cartographie et de suivi simultanément à l'aide d'une hypothèse planaire plus générale.
3. Du point de vue de la relocalisation du système SLAM, l'idée est de récupérer les endroits déjà passés par l'agent robot pour éliminer l'erreur d'estimation globale ou lorsque le robot est en état perdu. Nous avons proposé une structure de graphe avec des embedding binaire pour intégrer des informations spatiales et des formats de données hétérogènes tels que des images de profondeur, des informations sémantiques, même des résultats de deep learning etc. La méthode proposée permet aux systèmes robotiques SLAM de se relocaliser avec un taux de réussite plus élevé, même dans des conditions de différentes éclairage, météorologiques et saisonnières.

Publications Relatives à la Thèse

1. Xi Wang, Marc Christie, and Eric Marchand, « Multiple Layers of Contrasted Images for Robust Feature-Based Visual Tracking », *in*: *IEEE Int. Conf. on Image Processing (ICIP)*, 2018

2. Xi Wang, Marc Christie, and Eric Marchand, « Optimized Contrast Enhancements to Improve Robustness of Visual Tracking in a SLAM Relocalisation Context », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018
3. Xi Wang, Marc Christie, and Eric Marchand, « Relative Pose Estimation and Planar Reconstruction via Superpixel-Driven Multiple Homographies », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020
4. Xi Wang, Marc Christie, and Eric Marchand, « TT-SLAM: Dense Monocular SLAM for Planar Environments », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Xi'an, China, 2021
5. Xi Wang, Marc Christie, and Eric Marchand, « Binary Graph Descriptor for RobustRelocalization on Heterogeneous Data », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2022)

Chapter 3

Background on visual SLAM techniques

SLAM (Simultaneous Localization and Mapping) is one of the key techniques deployed in modern multi-sensor intelligent agents: robots, drones to portable augmented reality devices and autonomous driving vehicles. A standard SLAM system is designated to achieve localization tasks and environment recording via information acquired from sensors and computation algorithms in a parallel fashion. Visual SLAM (or vSLAM) is a specialization of SLAM where optical devices are exploited as the input sensors for its relatively low prices, easy access, high precision and rich information comparing to other sensors. However, to achieve the goal of localising in a three-dimensional world via a illumination sensor (digital camera), one requires to solve multiple problems such as:

1. imaging theory and projective geometry for building a physical model to reconstruct three-dimensional information from images;
2. general statistical, algorithmic, and mathematical techniques for improving precision, rapidity, and practicality to make the vSLAM tasks engineeringly feasible finally;
3. specific digital image processing and robotics vision tools for helping the vSLAM system confront more difficult environments or more complicated usage scenarios;

This chapter will introduce a background elaboration of vSLAM techniques: from camera imaging mechanism to back-end computation algorithms on the above aspects.

3.1 Three Dimensional Vision

SLAM (Simultaneous Localization and Mapping) systems hold a relatively long history compared with other computer vision and visionary robotics techniques. It tries to simultaneously answer the problem of robotics agent status estimation and environment reconstruction. No actual limits in types of sensors were set at the very beginning stage of SLAM techniques, though after a growing development and maturing fabrication industry of digital cameras, camera-based SLAM (*i.e.* vSLAM) is one of the leading solutions in terms of both research and technical aspects.

The very fundamental of SLAM techniques inherits from some geometric mathematical and physical theories, including three-dimensional rigid-body motion, perspective geometry, epipolar constraints, digital imaging models and optimizations techniques.

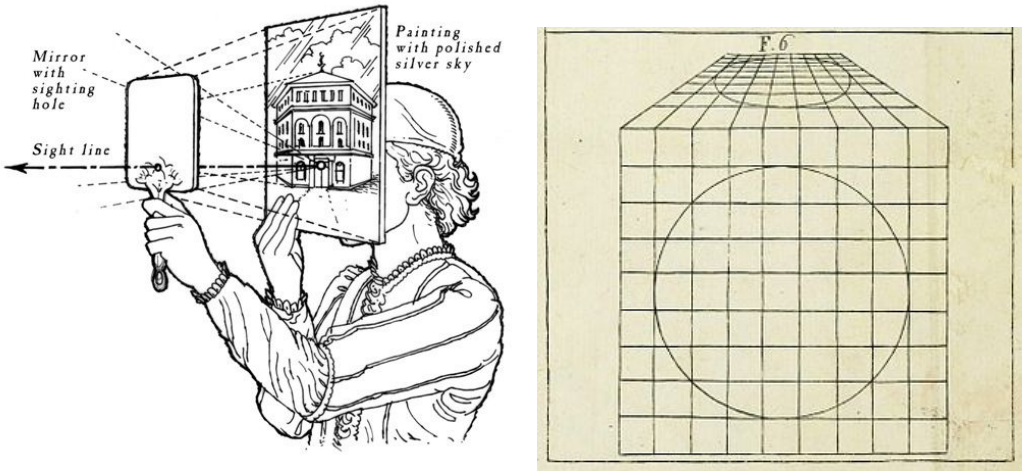
In this chapter, we try to draw a brief roadmap to describe all aforementioned backgrounds and concepts related to SLAM techniques as the main focus of this thesis.

3.1.1 A Brief History of Perspective Geometry

When Albrecht Dürer published his masterpiece *Instruction on Measurement* in 1525, he surely did not anticipate that he not only revolutionised the basic understanding and techniques of fine arts but also helped inspire a novel domain named projective geometry after centuries which fundamentally gave birth to the modern optic camera sensor and numerous innovations and applications in this very pedigree. The history of perspective vision and geometry can be traced back to the Italian Renaissance age, first participating in painting and architecture processing. Filippo Brunelleschi, a florentine architect, known by his major work: the dome of the Cathedral of Santa Maria del Fiore (the Duomo) in Florence, demonstrated one of his discoveries: a hole in the back of a painting and a parallel mirror in front of the paint. It helped Brunelleschi make sure of drawing a precise replica of his naked eyes' view (See Fig. 3.1a).

Inspired by Brunelleschi's experimental discoveries, his friend Leon Battista Alberti wrote the first general treatise on the laws of linear perspective, De Pictura, in 1435, illustrating one-point perspective in drawing (See Fig. 3.1b). After nearly a hundred years, Albrecht Dürer extended it to two-points perspective by reading formers' works.

Unlike Dürer addressed the problem of measurement by sheerly intuitive and empirical description: Girard Desargues, a French mathematician, published his *perspective theorem* in 1648. That was finally integrated into the group theory by German mathematician Felix



(a) A perspective checking device invented by Brunelleschi

(b) Figure from the 1804 edition of De Pittura

Figure 3.1 – (a) A device invented by Brunelleschi for verifying if the painting is identical to perspective observation by looking through a parallel mirror with peepholes on both mirror and drawing. (b) Figure from the 1804 edition of *De Pittura* written by Leon Battista Alberti shows the one-point perspective about the transformation from a circle to an ellipse.

Klein after various developments and complements.

Nowadays, in the computer vision and image domain, one usually relies on mathematics tools such as linear algebra and group theories for illustrating the geometric imaging model.

3.2 Projective Geometry

Unlike some three-dimensional sensors such as LIDAR (LIght Detection and Ranging) and other types depth sensors, a digital camera outputs a two-dimensional image per time, capturing projected information from a three-dimensional world, given the specific angle and camera position. Therefore, to recover the complete three-dimensional information, mathematical, physical and even optical models are required to support the objective of reconstructing projected images theoretically.

3.2.1 3-D Coordinate Frame and Rigid-Body Motion

Before delving into the mechanism of projection, we present basic mathematical tools for describing the nature and relation of objects in the three-dimensional world. A term *frame* $\mathcal{F}$ is utilized in which the coordinates of objects are able to be described mathematically. A frame contains an origin which indicates an initial position, such that all coordiantes in this frame are presented and anchor as the relative coordinates towards this very origin. We define two frames: $\mathcal{F}_c$ and $\mathcal{F}_w$, camera frame and world frame respectively for representing the frame attached to the camera center as well as the one to the object as we are interested by the relation and relative motion between these two objects. A three-dimensional point in the world frame $\mathcal{F}_w$ can be depicted as ${}^w\overline{\mathbf{X}} = ({}^wX, {}^wY, {}^wZ)^\top$ and similarly for camera frame $\mathcal{F}_c$: ${}^c\overline{\mathbf{X}} = ({}^cX, {}^cY, {}^cZ)^\top$. The relation between these two frames is called a transformation, *eg.* ${}^c\mathbf{T}_w$ can express a transform from world frame $\mathcal{F}_w$ to camera frame $\mathcal{F}_c$.

Considering the motion of one object, it requires to specify the trajectory of each point's transform if no other condition is given. For rigid objects, it is sufficient to describe the whole object's behaviour by only considering one arbitrary point and its coordinate frame instead of including every point of the object into the calculation.

A rigid-body displacement comprises two types of motion: a translational one and a rotational one, *eg.* moving an object from one place to another. In order to depict mathematically a displacement in the Cartesian system, it consists of a combination of these two motions:

$$ {}^c\overline{\mathbf{X}} = {}^c\mathbf{R}_w {}^w\overline{\mathbf{X}} + {}^c\mathbf{t}_w \tag{3.1} $$

where ${}^c\mathbf{R}_w$ and ${}^c\mathbf{t}_w$ are denoted as rotation matrix and translation vector of a rigid body motion bringing a point from world to camera sensor frame.

3.2.2 Homogeneous Representation

We represent three-dimensional points and the correspondent transform with the help of homogeneous coordinates, by adding a 1 at the end of the linear coordinate (*eg.* $\overline{\mathbf{X}} = (X, Y, Z)^\top$). The motivation of this affine design helps make coordinates transformation

(*eg.* ${}^{c}\mathbf{T}_{w}$) tractable by linear matrix multiplication though by adding little redundancy:

$$\begin{pmatrix} {}^{c}\overline{\mathbf{X}} \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{c}\mathbf{R}_{w} & {}^{c}\mathbf{t}_{w} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{w}\overline{\mathbf{X}} \\ 1 \end{pmatrix} \tag{3.2}$$

Therefore the transform we mentioned previously for converting of rigid-body motion holds an affine matrix of ${}^{c}\mathbf{T}_{w} \in \mathbb{R}^{4\times4}$:

$${}^{c}\mathbf{T}_{w} = \begin{pmatrix} {}^{c}\mathbf{R}_{w} & {}^{c}\mathbf{t}_{w} \\ \mathbf{0}_{3\times1} & 1 \end{pmatrix} \tag{3.3}$$

The matrix ${}^{c}\mathbf{T}_{w}$ belongs to the Special Euclidean Group $SE(3)$ whereas its rotational part ${}^{c}\mathbf{R}_{w}$ belongs to a Special Orthogonal Group $SO(3)$ embedded in $\mathbb{R}^{3\times3}$:

$${}^{c}\mathbf{R}_{w} \in SO(3) \quad \text{s.t.} \quad SO(3) = \left\{ {}^{c}\mathbf{R}_{w} \in \mathbb{R}^{3\times3} \mid {}^{c}\mathbf{R}_{w}{}^{\top} {}^{c}\mathbf{R}_{w} = \mathbf{I}_{3}, \det\left({}^{c}\mathbf{R}_{w}\right) = 1 \right\} \tag{3.4}$$

The Special Euclidean Group $SE(3)$ for describing rigid-body motions in three-dimensional world can be therefore expressed bya adding translational vector ${}^{c}\mathbf{t}_{w} \in \mathbb{R}^{3}$ together with the rotation matrix in homogeneous representation:

$$SE(3) = \left\{ {}^{c}\mathbf{T}_{w} = \begin{pmatrix} {}^{c}\mathbf{R}_{w} & {}^{c}\mathbf{t}_{w} \\ \mathbf{0}_{3\times1} & 1 \end{pmatrix} \mid {}^{c}\mathbf{R}_{w} \in SO(3), {}^{c}\mathbf{t}_{w} \in \mathbb{R}^{3} \right\} \tag{3.5}$$

Once the above notions are defined, we can simply transfrom a object from $\mathcal{F}_{w}$ to $\mathcal{F}_{c}$ by a matrix multiplication:

$${}^{c}\mathbf{X} = {}^{c}\mathbf{T}_{w}{}^{w}\mathbf{X} \tag{3.6}$$

3.2.3 Projective Geometry

A 2-Dimensional image is generated as a result of a projection from a 3-Dimensional world to a 2-Dimensional camera plane. The procedure of computing the projection of a 3-D point in camera frame $\mathcal{F}_{c}$: ${}^{c}\mathbf{X} = ({}^{c}X, {}^{c}Y, {}^{c}Z, 1)^{\top}$ and its corresponding projected on-image 2-D homogeneous coordinate: $\mathbf{x} = (x, y, 1)^{\top}$, where the on-image coordinates

(x, y) are normalised by the Z direction depth, is expressed as:

$$\begin{cases} x = \frac{^cX}{^cZ} \\ y = \frac{^cY}{^cZ} \end{cases} \tag{3.7}$$

When the center of the image coincides with the center of the coordinate (*i.e.* optic center), a shifting and scaling operation yield image pixel order $\mathbf{x}_p = (u, v)$:

$$\begin{cases} u = u_c + p_x x \\ v = v_c + p_y y \end{cases} \tag{3.8}$$

where p_x and p_y are the scaling ratio to pixel on x and y direction, and u_c and v_c are the principal point on the image plane corresponding to the optic center. By combining the above equations and notions, we finally reach a full description of the projection from 3-D points to 2-D image pixels:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} p_x & 0 & u_c \\ 0 & p_y & v_c \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\Pi} \begin{pmatrix} ^cX \\ ^cY \\ ^cZ \\ 1 \end{pmatrix}. \tag{3.9}$$

where we call the $\mathbf{K}$ the intrinsic matrix or calibration matrix and Π the extrinsic matrix, to rewrite it in a compact fashion for representing a transform from $\mathcal{F}_w$ to $\mathcal{F}_c$, converting a homogeneous spatial coordinate ${}^w\mathbf{X}$ to a homogeneous image coordinate $\mathbf{x}_p$:

$$\mathbf{x}_p = \mathbf{K}\mathbf{\Pi}^c\mathbf{T}_w{}^w\mathbf{X} \tag{3.10}$$

3.2.4 Geometric Constraints

Given that the projection is described by a specific mathematical model, geometric constraints are the relations of the pixel points between the multiple views and to the 3-D model. Multiple constraints exist when giving the different conditions. This section focuses on the these geometric relations and their mathematical representations.

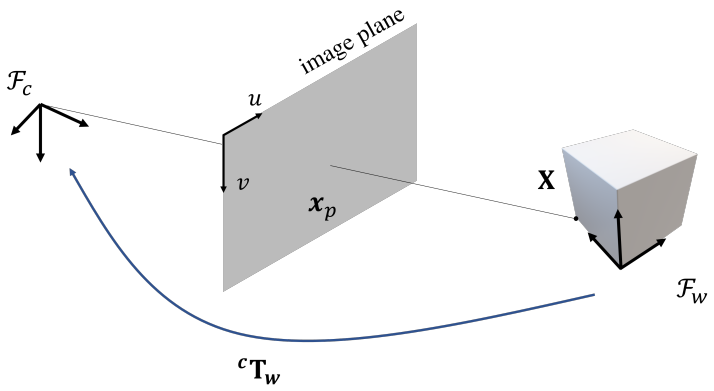


Figure 3.2 – Projecting an object from world frame to image plane represented in camera frame

Epipolar Constraint and Essential Matrix

Epipolar geometry studies the geometric constraints between two views projected from the same 3-D model. Given two images taken from two distinct views of the same scene (the assumption is that the scene is static and illumination invariant), under the condition that the camera is calibrated (*i.e.* the $\mathbf{K}$ matrix is known in equation 3.10), with previously mentioned homogeneous image and spatial coordinates: $\mathbf{x}$ and $\mathbf{X}$, one can draw the relation:

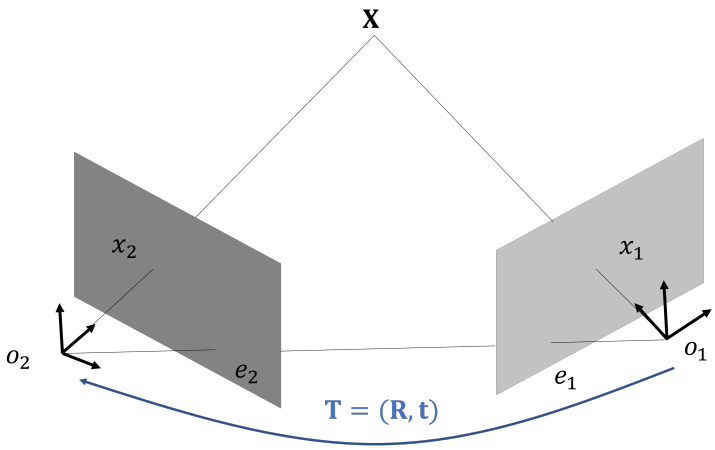


Figure 3.3 – A geometric example of epipolar constraint, in which the $\boldsymbol{x}_1, \boldsymbol{x}_2$ are homogeneous image coordinates of $\boldsymbol{X}$ a 3-D point $\boldsymbol{X}$. The Euclidean transform between two cameras is defined by $(\mathbf{R}, \mathbf{t}) \in SE(3)$. The intersections of the translational vector $\mathbf{t}$ and two image planes are called epipoles: (e_1, e_2).

$$\lambda \boldsymbol{x} = \boldsymbol{\Pi X} \tag{3.11}$$

where the $\boldsymbol{\Pi} = [I, 0]$ and $\lambda \in \mathbb{R}_+$ is an unknown scale. Applying the above relation with the equation of rigid-body motion (Eq. 3.1), one can describe the equation of a homogeneous image point after a $SE(3)$ transform on spatial coordinate from homogeneous coordinate $\boldsymbol{x}_1$ to another one $\boldsymbol{x}_2$:

$$\lambda_2 \boldsymbol{x}_2 = \mathbf{R}\lambda_1 \boldsymbol{x}_1 + \mathbf{t} \tag{3.12}$$

Premultiply an $[\mathbf{t}]_\times$ on both sides of preceding equation ($[\mathbf{t}]_\times \mathbf{x} = \mathbf{t} \times \mathbf{x}$ a cross-product)

$$\lambda_2 [\mathbf{t}]_\times \boldsymbol{x}_2 = [\mathbf{t}]_\times \mathbf{R}\lambda_1 \boldsymbol{x}_1 \tag{3.13}$$

Since the cross-product $[\mathbf{t}]_\times \boldsymbol{x_2}$ gives a perpendicular direction to both vectors, an inner-product with $\boldsymbol{x_2}$ is then exploited for finally eliminating the ambiguity of scale factor λ, for that $\boldsymbol{x}_2^\top [\mathbf{t}]_\times \mathbf{R}\lambda_1 \boldsymbol{x}_1$ is zero. Therefore epipolar constraint develops to:

$$\boldsymbol{x}_2^\top [\mathbf{t}]_\times \mathbf{R} \boldsymbol{x}_1 = 0 \tag{3.14}$$

we often replace the multiplication with a matrix $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$ termed essential matrix.

Planar Constraint and Homography Matrix

If we apply a stronger assumption on 3-D coordinates such as all the observed points are coplanar (*i.e.* coplanarity), the above equation will share an extra constraint rather than an epipolar constraint. Let $N = (n_1, n_2, n_2)^T \in \mathbb{S}^2$ as the unit normal vector of a plane P in the coordinate world of the first camera frame, and $d \in \mathbb{R}_+$ denotes the distance from the plane to the origin of the first camera frame:

$$N^T \boldsymbol{X}_1 = n_1 X + n_2 Y + n_3 Z = d \quad \Leftrightarrow \quad \frac{1}{d} N^T \boldsymbol{X}_1 = 1, \quad \forall \boldsymbol{X}_1 \in P \tag{3.15}$$

We follow the same method via rigid-body motion relation:

$$\boldsymbol{X}_2 = \mathbf{R}\boldsymbol{X}_1 + \mathbf{t} = \mathbf{R}\boldsymbol{X}_1 + \mathbf{t}\frac{1}{d} N^\top \boldsymbol{X}_1 = \left(\mathbf{R} + \frac{1}{d}\mathbf{t}N^\top\right) \boldsymbol{X}_1 \tag{3.16}$$

Homography matrix is then described as:

$$\boldsymbol{H} \doteq \mathbf{R} + \frac{1}{d}\mathbf{t}N^{\top} \in \mathbf{R}^{3\times 3} \quad \text{or simply} \quad \boldsymbol{X}_2 = \boldsymbol{H}\boldsymbol{X}_1 \tag{3.17}$$

prejecting on homogeneous image coordinates by $\lambda_i \boldsymbol{x}_i = \boldsymbol{X}_i$, a relationship between images taken from two views is defined as follows:

$$\lambda_2 \boldsymbol{x}_2 = \boldsymbol{H}\lambda_1 \boldsymbol{x}_1 \quad \Leftrightarrow \quad \boldsymbol{x}_2 \sim \boldsymbol{H}\boldsymbol{x}_1 \tag{3.18}$$

where $\sim$ represents equality up to a scalar factor.

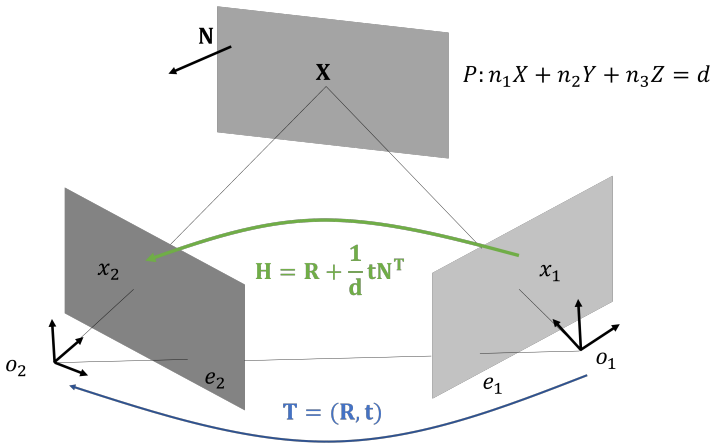


Figure 3.4 – A geometric example of the homography constraint, in which the $\boldsymbol{x}_1, \boldsymbol{x}_2$ are homogeneous image coordinates of $\boldsymbol{X}$ a 3-D point lying in a plane P.

3.3 Nonlinear Optimization in SLAM

As we introduced in the previous sections, many computer vision tasks rely on low-level landmarks to build geometric constraints for reconstructing 3-D information. As this nonlinear procedure is often overdetermined and noise influenced, probabilistic modelling and optimization techniques are applied to help improve the precision and the robustness of SLAM and stereo vision applications. This section concentrates on nonlinear optimization from definition to the classic applications in the SLAM context.

The general definition of nonlinear optimization is about successively minimizing a nonlinear function by updating objective parameters *s.t.* the given cost function decreases at each step. In the SLAM context, the Gauss-Newton family methods are generally

regarded as the most representative algorithms for achieving the tasks such as constraint estimation, decomposition, landmarks alignment and the Bundle Adjustment, which we will discuss later.

Take the Gauss-Newton method as an example, given a residual function r of n objective parameters as a vector $\boldsymbol{\beta} = (\beta_1, ..., \beta_n)$. In SLAM-like context, the residual function are usually defined as the euclidean error of landmarks projected on image views. The Gauss–Newton algorithm iteratively finds to update the value of the parameters β which minimizes the sum of squares of residual generated $r(\boldsymbol{\beta})$:

$$S(\boldsymbol{\beta}) = \boldsymbol{r}(\boldsymbol{\beta})^2 \tag{3.19}$$

the update of the parameters $\boldsymbol{\beta}^{(s+1)}$ based on previous step $\boldsymbol{\beta}^{(s)}$ is given:

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - \left(\mathbf{J}_\mathbf{r}^\top \mathbf{J}_\mathbf{r}\right)^{-1} \mathbf{J}_\mathbf{r}^\top \mathbf{r}\left(\boldsymbol{\beta}^{(s)}\right) \tag{3.20}$$

where the $\mathbf{J}$ is Jacobian, the matrix of partial derivatives on the residual function (*i.e.* obejctive function) w.r.t parameters to optimize:

$$(\mathbf{J}_\mathbf{r})_i = \frac{\partial r\left(\boldsymbol{\beta}^{(s)}\right)}{\partial \beta_i} \tag{3.21}$$

$i = 1, ..., n$ represents each dimension of interested parameter vetor $\boldsymbol{\beta}$.

3.3.1 Jacobian Matrix

Specifically in the perspective geometry we presented above, the partial derivative of the projection equation (Eq. 3.7) w.r.t x and y direction on image space:

$$\begin{cases} \dot{x} = \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z \end{cases} \tag{3.22}$$

Then relating the 3-D velocity on the translational and rotational velocity with ω_c computed from Rodrigez formula:

$$\dot{\mathbf{X}} = -\boldsymbol{v}_c - \omega_c \times \mathbf{X} \Leftrightarrow \begin{cases} \dot{X} = -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = -v_y - \omega_z X + \omega_x Z \\ \dot{Z} = -v_z - \omega_x Y + \omega_y X \end{cases} \tag{3.23}$$

Combing above two equations, we have:

$$\begin{cases} \dot{x} = -v_x/Z + xv_z/Z + xy\omega_x - (1+x^2)\,\omega_y + \gamma\omega_z \\ \dot{y} = -v_y/Z + yv_z/Z + (1+y^2)\,\omega_x - xy\omega_y - x\omega_z \end{cases} \tag{3.24}$$

Finally the Jacobian of projective model holds its form as, with each column representing direction on $\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z, \boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\theta}_z$ s.t. the $\mathbf{q}$ is a minimal representation of the transform $\mathbf{T} \in SE(3)$, $\mathbf{q} = ({}^c\mathbf{t}_w, {}^c\boldsymbol{\theta}_\mathbf{w}) \in \mathbb{R}^6$:

$$\mathbf{J} = \begin{pmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \tag{3.25}$$

3.3.2 Bundle Adjustment

We introduced means to estimate the Essential matrix from two different views in previous sections. However, it is not always easy to compute a common geometric constraint for arbitrary different view angles when dealing with sequential information. The standard solution lies in designing a so-called Bundle Adjustment system for minimising, in common, a sequence of information taken from different vantages and time. Under the scenario of points, Bundle Adjustment tries to computes an optimal solution under the constraint of all the observed points from each view and different time.

Denoting $\mathbf{q}$ a minimal representation of the transform, Bundle Adjustment holds the general form of a Maximum Likelihood Estimator for each data in the system, including observed landmarks and camera pose at each temporal positions:

$$\left([\hat{\mathbf{q}}]_t, [{}^w\hat{\mathbf{X}}]_n\right) = \underset{\left([\mathbf{q}]_t, [{}^w\mathbf{X}]_n\right)}{\arg\min} \sum_{j=1}^{\mathrm{t}} \sum_{i=1}^{\mathrm{n}} d\left(\mathbf{x}_i, \mathbf{K}\Pi^j \mathbf{T}_w \, {}^w\mathbf{X}_i\right) \tag{3.26}$$

with $[\hat{\mathbf{q}}]_t$ and $[{}^w\hat{\mathbf{X}}]_n$ a sequence of t camera poses and n observed 3-D landmarks positions in world frame respectively. The optimization framework basically searches for the best landmarks position as well as the camera poses such that the reprojection errors on 2-D image coordinates can be minimized.

Usually, the problem of Bundle Adjustment fits well with the nonlinear optimization framework. However, given the sparse nature of the Bundle Adjustment (BA) problem (not all landmarks are observed by all cameras), many techniques were proposed for accelerating and simplifying this procedure *eg.* : graph-BA, sparse matrix compression, marginalization, etc. As those topics are out of this chapter's scope, we will not go into

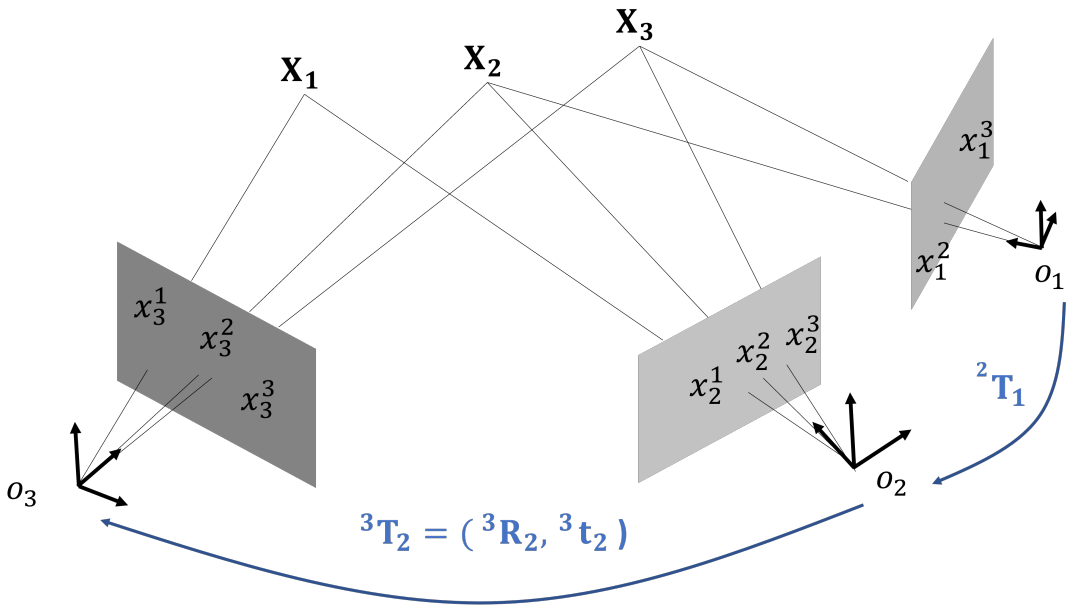


Figure 3.5 – Bundle Adjustment, in general, is defined as a nonlinear minimization on a sequence of multiple views and landmarks w.r.t to each camera pose and 3-D position of landmarks. An example is given here for three camera poses and three landmarks.

further details.

3.4 Image Representation and Image Processing

In the digital image world, the necessary discretization of real image plays a nonignorable role in numerically generating, storing and processing images. In this section, we are going through the basic concepts of digital image representation and image processing tools.

3.4.1 Images Representation

An image is, under the context of numerical camera photos, a 2-D brightness array (without considering color imaging and Bayer filter as they are out of scope of this thesis). A general representation of this *2-D brightness array* can be regarded as a map $\mathbf{I}$, defined on a compact region of a 2-D surface, often rectangular due to the equipped photon diode sensors (CMOS or CCD) and similar photographic mediums, containing positive values

of real or discrete numbers:

$$I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+; \quad (u, v) \mapsto I(u, v) \tag{3.27}$$

In the case in which we use 8-bit for storing brightness array value, the intensity of an image should fall into the interval of $\mathbf{I}(u, v) \in [0, 255]$. As far as this thesis is concerned, we adopt this definition by default if no further specifications are given.

3.4.2 Contrast and Histogram

A very primitive concept on digital images is the intensity histogram. It is a type of histogram showing the distribution of intensity levels w.r.t their quantity in a given image area. By manipulating the histogram, *eg.* histogram equalization or contrast stretching, one is able to adjust the contrast level of a digital image such that some compressed areas (whether too dark or too bright) are more visible (See Fig. 3.6).

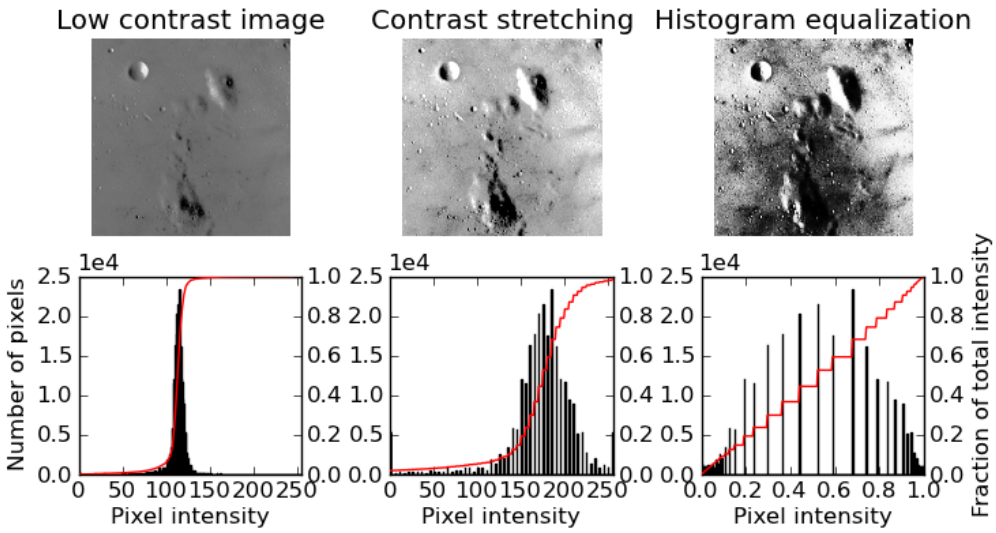


Figure 3.6 – An example of contrast streching and histogram equalization on a low-contrast image. Images are shown at the first row and histograms are shown in the second, with red curve representing the accumulated histogram.

3.4.3 Keypoints: Extractor and Descriptor

In computer vision, *keypoints* are an essential component of many algorithms and has being widely exploited in various applications, *eg.* : image stitching, image retrieval,

classification and of course, SLAM techniques.

Keypoints are local visual features the on-image 2D points showing significant contrast level. It helps identify remarquable regions in an image robust to environmental changes *eg.* from different vantages, scales, orientations or lighting conditions. In order to be tracked from different conditions properly, we need two modules: i) *extractor*, to extract higher contrast regions and ii) *descriptor*, to register and retrieve the found regions from another scene.

Extractors

An extractor is utilised to localise worth-tracking positions, *i.e.* points with higher local contrast, as strong contrast signifies good robustness against noises and environmental changes. Accordingly, standard methods are built on this contrast detecting concept, such as Harris corner [51] and Shi–Tomasi corner [56]. Harris corner works directly on the differential of the corner score concerning x and y directions. Instead of using pixel differentials to depict the form of the contrast, SIFT [68] detector uses the Difference of Gaussian (DoG) on downsampling scale image pyramids for extracting blob features across various scaling levels of the original image. SIFT detector assures well the detection and description afterwards when compared with others, but its computational cost remains a drawback.

On the contrary, in scenarios like real-time SLAMs and motion detections where people seek higher detection speed, FAST [98] extractor made its success. It relies on accelerated segment test and decision tree trained for balancing high-speed detection and relative acceptable accuracy and repeatability. FAST detector uses a 16 pixels circle (a Bresenham circle of radius equals 3 pixels) to localise and classify good candidates (See Fig. 3.7).

The principle consists in finding N contiguous pixels which demonstrate collective superior or inferior intensity level w.r.t the centre point. A threshold value is also used to increase the robustness against noises and illumination changes. Conditions are expressed as:

$$\begin{cases} \forall x \in S, I_x < I_p - t \\ \forall x \in S, I_x > I_p + t \end{cases} \tag{3.28}$$

where I_x are points on the circle and I_p is the candidates point at the centre.

The high-speed test is achieved by first checking on 4 example pixels (when $N = 12$), in the paper they proposed pixel 1, 9, 5 and 13. A specific order of checking and a quick

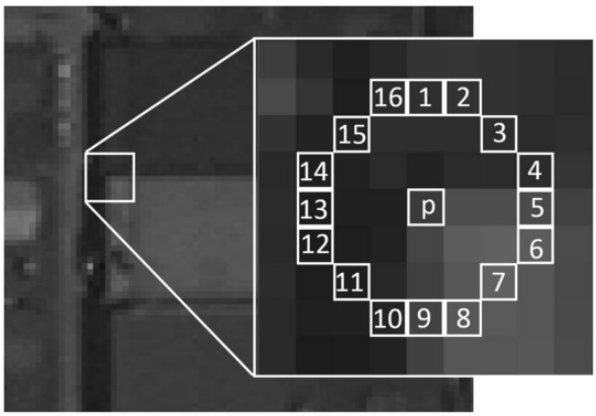


Figure 3.7 – An example of Fast detecting Bresenham circle of 16 pixels and their indexes.

abundance of unqualified candidates shorten the execution time. A decision tree-based machine learning process is later introduced for generalising the case when N is not equal to 12 and refining its decision order from a data-driven aspect.

For assessing an extractor's quality, an evaluation metric termed as repeatability is proposed: The definition of repeatability is the capacity of refinding the same keypoints from another view in terms of their on-image positions. More details will be elaborated in later sections.

Descriptors

In opposition with the repeatability test, where the ground truth correspondences of found keypoints are already known, real-life applications also require a specific module for describing, discriminating and corresponding same keypoints yielded by extractors under different conditions. In other words, one needs to know the correspondences (*i.e.* matching results) of keypoints taken from different views.

One of the simplest descriptors is to use the image patch directly and computing their photometric errors as SSD (Sum of Squared Differences) or other metrics for searching for the best matching result between two keypoint sets.

Many descriptors are crafted jointly to ensure good relevant characteristics of certain detectors, *eg.* SIFT detector utilizes a histogram of image gradient magnitudes and orientations, then is assigned to the keypoint orientation computed in the previous step for dealing with the rotation problem.

BRIEF (Binary Robust Independent Elementary Features) descriptor seeks a different angle to deal with the matching problem: it exploits random numerical "fingerprints" to

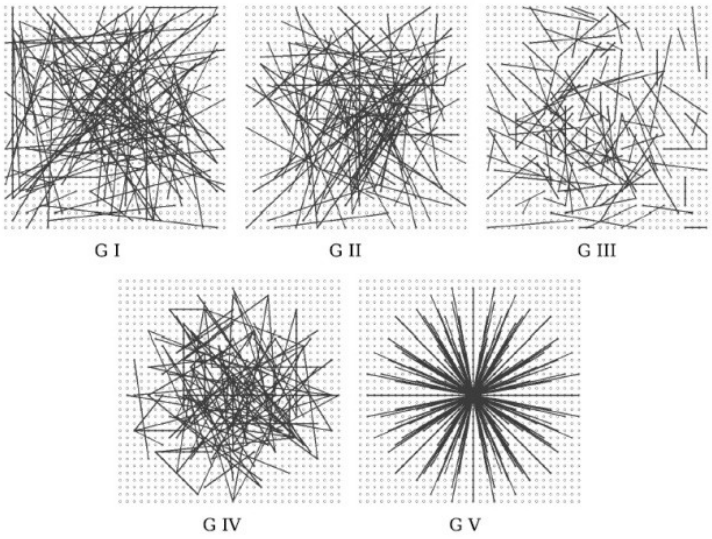


Figure 3.8 – Examples of random fingerprints used by BRIEF descriptor

discriminate different image patches. After smoothing the image patch, a straightforward binarization on random generated or predefined fingerprints pairs gives a certain digit of bits for describing it:

$$\begin{cases} 1, & p(x) < p(y) \\ 0, & p(x) \geq p(y) \end{cases} \tag{3.29}$$

where $p(x)$ and $p(y)$ are pixel intensities on the two ends of a fingerprint pair.

For matching two keypoints, the common solution is to apply Hamming distance on two generated binary bits, similar to an XOR operation, then counting the number of bits equal to 1. The advantage of BRIEF descriptor lies in its fast speed, as many CPUs are equipped with hardware acceleration on binary operations. Though it faces some difficulties when dealing with the rotation perturbations, some amendments were proposed, such as generating each BRIEF descriptor for all discretized orientations (*eg.* 64 orientations to split 360 degrees). BRIEF and its derivative techniques (rBRIEF used in ORB [100] detector) do appear in a wide range of applications where speed is a requirement.

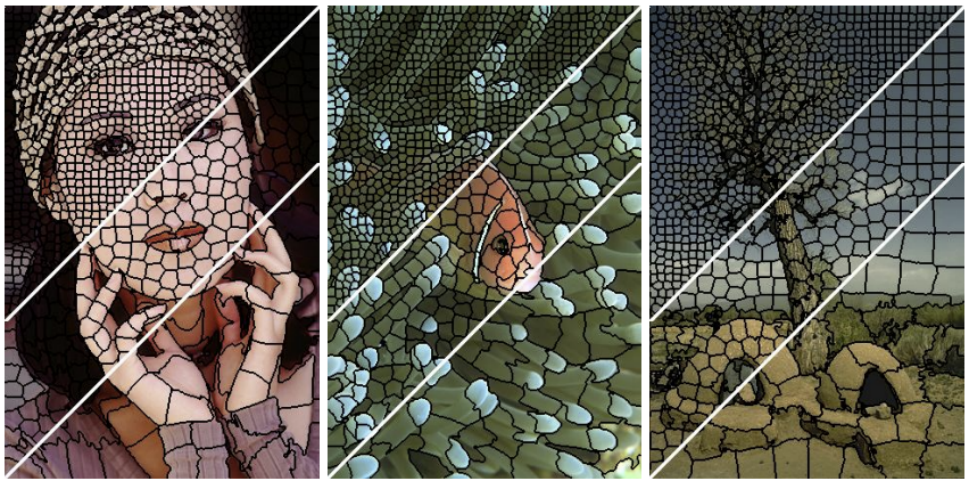


Figure 3.9 – Examples of SLIC superpixel segmentation on different size parameters. SLIC can output compact and regular superpixels of required size parameter with relative low overhead, figure from SLIC paper [1].

3.4.4 Superpixel and Segmentations

As introduced in the previous sections, people seek to capture and retrieve information from the 2-D images at different scales: local scale suggests the creation of the keypoints and global idea leads to the applications of histogram. A middle scale also exists and focusing on utilizing a mid-level group of pixels to describe the desired information. Some researches then propose to exploit segmentation methods to extract clusterings or regions which shares a level of similarity in terms of chromaticality, luminance, spatiality or even semantics.

Depending on different criteria, different segmentation methods are proposed including, color driven image segmentation, semantic segmentation, human detection and segmentation etc. Superpixel techniques, a segmentation approach, focus on the local spatial and chromatic similarity and primitive geometries such as compactness. As an example, SLIC [1, 96] is able to generate compact yet regular segmentation regions with low computational cost (See Fig. 3.9). Many applications of superpixels can be found in mordern computer vision and robotics vision domains such as SLAM [24, 23], visual tracking [125], etc.

3.5 Conclusion

In this chapter, we recalled some fundamental concepts and mathematical tools, from rudiments of rigid-body motion to the mechanism of imaging model, crossing domains of three-dimensional vision, nonlinear optimization and image processing. It resumes from the history of perspective in fine arts to the definitions, elements and properties of projective geometry. Since the SLAM problem is also considered as an optimization and estimation problem, nonlinear optimization tools play an essential role in organising and estimating camera poses and landmarks. In the last part, we mentioned some practical numerical image concepts and image processing tools for visual tracking, pose estimating and SLAM, as they will be largely discussed in the following chapters.

Chapter 4

Related works on robust visual SLAM techniques

SLAM, specifically visual SLAM, is a challenging and largely addressed problem in the computer vision and robotics field. The definition of the SLAM techniques includes a large range of different directions: From filter-based methods to neural networks SLAMs. In this thesis, we will mainly focus our topic on the sparse keypoints-based visual SLAMs and the robustness problem during each step of SLAM system. In this section, relevant related works will be discussed in terms of their motivation, technical structure, advantages and disadvantages, wishing to depicting the current situation and the development of our interested problems.

4.1 The Development of vSLAM Systems

Visual SLAM also called vSLAM, is a direction of SLAM techniques where optical devices are exploited as the input sensors for its accessibility, low price and rich information. It consists of achieving the goal of localizing in a unknown environment and building a three-dimensional map simultaneously.

Mainstream vSLAM methods divagated multiple times along its development, in order to improve the system in response to key challenges: i) the precision of estimation; ii) the computational cost when confronting long sequence; iii) the robustness against enviroment and noisy inputs.

4.1.1 Filter-based SLAMs

Early SLAM works mostly concentrated on odometry problems and using laser and ultrasonic sensors as perception input. The paradigm of solving SLAM problems starts on probabilistic estimation models and tends to exploit corresponding filter-based toolboxes: *eg.* Bayensien-based filters (Kalman filters and particle filters), MAP (Maximum a Posteriori) estimation, etc.

First monocular vSLAM was invented by Davison et al. in 2003, namely MonoSLAM [30]. It brings the image features into the visibility of SLAM research. The method consists of extracting local image patches as landmarks in the map and updating the features' depth for preserving frame-to-frame matching and 3D reconstruction. This representative filter-based vSLAM work exploits the EKF (Extended Kalman Filter) for treating a state vector of 6 DoF (Degree-of-Freedom) and 3-D landmark locations for achieving the estimation and mapping. However, the main problem of EKF-based SLAMs suffer from two aspects: i) outlier rejection as the image features may include erroneous inputs. ii) high computational cost w.r.t the continuously growing map (at cubic rate). Usual countermeasures include submap filtering, local-global map division and graph map structures.

4.1.2 Optimization-based SLAM

Optimization-based vSLAM techniques generally comprise two modules: One performs the data acquisition from sensors including data filtering, association with the local map, and even outliers rejections, usually termed as frontend or tracking module. A rough relative pose estimation is output in this procedure. Local tracking can be achieved

under the asynchronously updated map for outputting the current sensor pose: standard methods include PnP [37] (Perspective-n-Points) and ICP [7] (Iterative Closest Points) depending on the nature of the sensors and format of acquired data (2-D or 3-D).

The other module takes care of multiple poses refining, global constraining and drifting elimination, *i.e.* local mapping and bundle adjustment. This module is also called mapping or backend under some contexts. As the computational cost is usually higher than the frontend tracking, in many applications the update frequence is lower and asynchronous to the real-time tracking module. Some proposed to add loop detection and relocalization system in the backend for building extra constraints during the optimization process.

A widely considered categorisation is to differentiate frontend parts, though the backend part should adjust accordingly between: (i) sparse methods or feature-based methods: relying on sparsely extracted keypoint-features or low-level landmarks; (ii) direct methods or dense methods: dissimilar to feature-based methods selecting and extracting low-level features, dense methods utilize the whole acquired image.

Feature-Based SLAM

As previously mentioned, feature-based SLAMs focus on extracting low-level features (often keypoints we introduced in the background chapter), tend to minimise the reprojection error, defined as the on-image reprojected distance of visual features from different vantages.

PTAM (Parallel Tracking And Mapping): The first intuition of feature-based optimization driven SLAM is to solve the scale growing problem of EKF-SLAMs. PTAM [62] decouples the tracking and mapping into two parallel threads: frontend tracking part and backend mapping part, namely Bundle Adjustment (BA) optimization, which incorporates camera poses together with landmarks positions. PTAM is the first paper to exploit BA and this parallel structure to free computational cost and bring monocular SLAM into the real-time era. The improvement consists in stating that tracking tasks are lighter and easier to compute without refining positions of the landmarks simultaneously. The initialisation of the PTAM system starts with a five-point algorithm for estimating initial poses and keypoints. FAST detector [98] and patch searching scheme are utilised during the tracking procedure. During the mapping part, instead of applying optimization on every frame, the idea of keyframe is exploited in PTAM paper to acheive enough disparity (also called translational distance or baseline) for computing triangulation and constructing geometric constraints. Another obvious advantage of keyframes is that it lowers the

computational burden as the cost is proportional to the number of considered frames. Local BA is also introduced in this chapter to optimize neighbour keyframes and a local map, contrary to global BA that includes all keyframes and landmarks. Last but not least, this is the first time a relocalization method is employed in a vSLAM system. A randomised tree-based feature classifier is designed for searching the most similar keyframe in the dataset against the acquired input. To find and correct drifting errors accumulated during the tracking and mapping along the time.

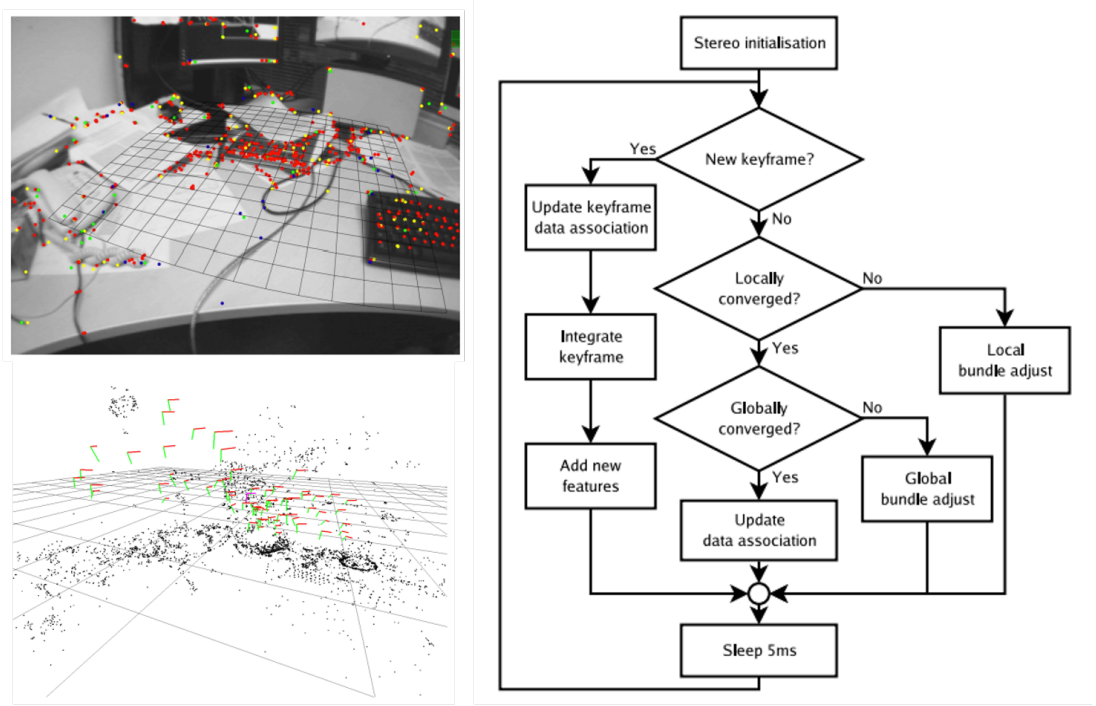


Figure 4.1 – An example of PTAM system on tracking (left-up) and mapping (left-bottom) ends with its pipeline (right), left-up subfigures shows the tracking view of PTAM and corresponding keypoints in a generated point cloud map. The three-dimensional view of the generated point cloud map and computed poses are demonstrated on the left-bottom side. The right side displays the pipeline of the PTAM system (Figure from [62]).

In general, PTAM can be regarded as one exemplary (one may assert the most important one) of the modern optimization-driven feature-based SLAMs: it introduces many essential concepts such as keyframes, local/global BA, and relocalization. All these elements and modules have been extended in almost all vSLAM systems proposed afterwards.

Compared to MonoSLAM and similar EKF-based SLAMs, PTAM shows an excellent ability to create a large map and simultaneously perform real-time tracking. A mobile phone version is further developed thanks to its compatibility and low computational cost.

Some extended versions are also introduced, especially for completing the sparse mapping to dense reconstruction. One good example is to use superpixel techniques on a PTAM system for dense mapping problem [24].

ORB-SLAM: Following the idea of PTAM in terms of dual threads structure, ORB-SLAM [77] improves the performance and robustness by a series of improvements: First, instead of using FAST and patch matching for image-to-image tracking, ORB-SLAM proposes to utilise ORB [100] features. ORB feature is a combination of oFAST detector, and rBRIEF descriptor. The difference to their original versions (FAST [98] and BRIEF [17]) are the improvements on the rotational correspondence. Besides its high-speed performance on extraction and matching tasks, a loop-closure system in the ORB-SLAM is specially designed for ORB-like descriptors: DBoW [44]. DBoW exploits binary descriptors to build a bag-of-words (BoW) system, via reusing an offline trained word dictionary, DBoW is able to retrieve similar candidates satisfying both visual feature consensus and geometric pose validation.

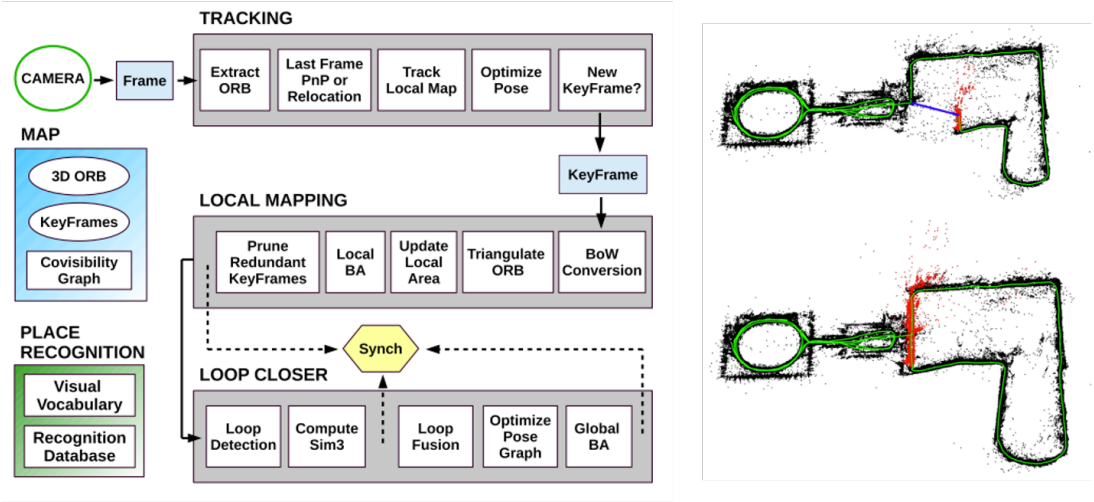


Figure 4.2 – The pipeline and an example of loop-closure executed by ORB-SLAM [77]). In the left side shows the pipeline of ORB-SLAM, three threads represent tracking, local mapping and loop closer respectively (Figure from [77])

ORB-SLAM demonstrated a stat-of-the-art level of relocalization performance. Once a suitable candidate loop-closure is found, global BA is launched to eliminate accumulated drifting errors (see Fig. 4.2). In addition, ORB-SLAM also brings other innovations into the feature-based SLAM. Compared to PTAM optimizing 6 DoF of camera pose, ORB-SLAM computes $Sim3$ ($SE(3)$ and a similarity scale, 7 DoF in total) like in the LSD-SLAM [34] for also incorporating the scale ambiguity due to monocular input. The

second innovation lies in keyframe insertion and removal policies: ORB-SLAM presents some empirical rules on adding and pruning keyframes for two motivations: i) constraining the total number of keyframes such that the system can run in real-time under larger-scale environments; ii) avoiding importing erroneous keyframes into the system, which improves the robustness against environmental noises such as changing illumination and moving objects.

As one milestone of feature-based SLAM, deriving from the basic structure of PTAM, ORB-SLAM can be seen as one of the most engineeringly ready monocular vSLAM today. It shows high precision, relative low computational cost, and can be extended with more sensors *eg.* RGB-D camera (ORB-SLAM2 [78]) and Inertial Measurement Units (ORB-SLAM3 [18]).

Direct SLAM

In contrast to feature-based methods, direct SLAM, also called the dense method, relies on aligning on-pixel similarity (*eg.* photometric error) between two images over their relative camera pose. Its most significant advantage is recovering a three-dimensional map densely with much better semantic meaning than the sparse point cloud generated by feature-based methods. Whereas the potential risk is obvious too, direct methods usually struggle in the dynamic and noisy cases as a result of its image-to-image alignment nature and the violated brightness constancy assumption of the scene.

DTAM (Dense Tracking and Mapping in Real-Time): DTAM [80] was proposed in 2011, a fully direct SLAM method. Its tracking part is performed by a direct registration between the current input image and reprojected 3D map reconstructed previously with the help of its implementation on GPU. The mapping part is achieved by multi-baseline stereo and optimized by considering the space continuity as regularisation. Finally, the initialisation of DTAM is similar to PTAM, by using a stereo measurement. In terms of robustness, DTAM sets photometric error thresholds during the optimization process to robustify the system, though authors indicate that the system is genuinely vulnerable against real-world global illumination changes.

LSD-SLAM (Large-Scale Direct Monocular SLAM): LSD-SLAM [34] is another milestone in the category of direct methods (see its pipeline in Fig. 4.3). Instead of using all pixels information (DTAM), LSD-SLAM maps only the areas with higher gradient intensity. The tracking is done by searching for optimal camera pose from a reconstructed KF by the Gauss-Newton method with variance-normalised photometric

error. The second difference to DTAM is that LSD-SLAM gives random values as initial depth guess for each pixel and expects them to be optimized with photometric consistency. The loop-closure module on LSD-SLAM relies on an appearance-based image retrieval algorithm: FAB-MAP [26]. $Sim3$ pose-graph optimization is porposed in this work too instead of rigid-body motion $SE(3)$ group.

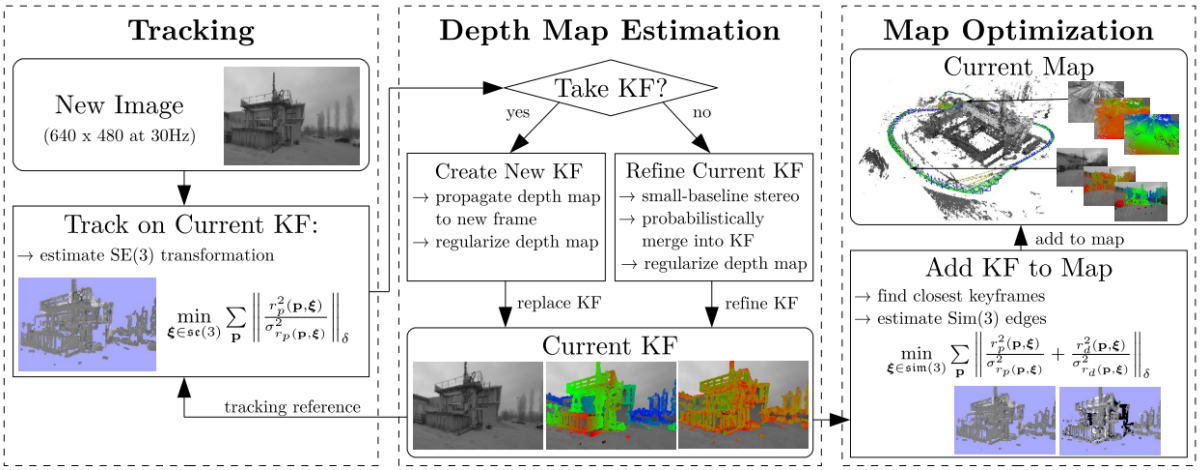
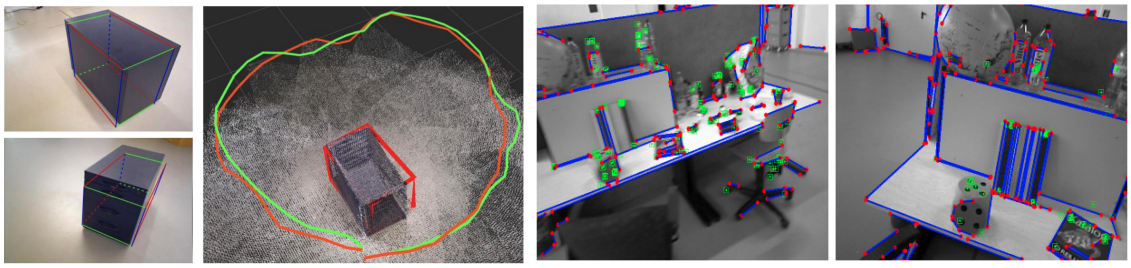


Figure 4.3 – The pipeline of direct method LSD-SLAM, the tracking is performed on high gradient intensities, and the mapping part relies on estimation of $Sim3$ edges in a pose-graph optimization. LSD-SLAM has a loop-closure module via appearance-based image retrieval method (Figure from [34])

Hybrid vSLAM

Readers may find from previous introductions and summaries that a tradeoff actually exists between the feature-based SLAMs and the dense SLAMs. Feature-based SLAMs usually claim higher precision, better computational efficiency and good robustness against noises thanks to the characteristics of the handcrafted low-level features and their auxiliary toolboxes *eg.* bag-of-words, matching techniques, RANSAC, etc. However, the point cloud generated from feature-based SLAMs shows a level of insufficiency in completeness and semantic meaning to be utilise and perceived by human beings. Direct methods answer the demand yet sometimes hindered by the illuminative changes. A question shall be asked: is there any way to combine their advantages and create precise tracking and dense mapping, simultaneously?

Some hybrid SLAM systems tries to address this third-way. Besides point landmarks and pixels, one can extract geometric primitives from 2D images. Examples include line,



(a) CubeSLAM (b) PL-SLAM

Figure 4.4 – In subfig (a), CubeSLAM utilises object detecting network to extract cube shapes from images (Figure from [134]). In subfig (b), PL-SLAM detects line descriptors for tracking and pose estimation (Figure from [88]).

plane and more complex shapes such as cube and sphere. A standard structure of these hybrid SLAM systems with geometric primitives is analogous to the feature-based SLAM structure with a dense mapping module. The difficulty is that geometric primitives lack a good mathematical representation during the optimization module, many choose to apply a simple linear combination on the BA errors.

Yang et al. [134] exploit deep-learning technique to abstract cube shapes and design a geometric error measure for the BA procedure. An example of using line features is PL-SLAM [88]. It extends ORB-SLAM with LSD line detecting [124] version and makes tracking, matching, and corresponding BA modifications to adapt line features into the original ORB-SLAM structure. Higher precision, specifically in textureless scenes, is achieved with a map composed of lines and points which provides a higher semantic view, than traditional point cloud based approaches.

Unlike other primitives as line and shape, the plane does have a stand-alone mathematical representation in projective geometry, namely the homography constraint we introduced in the background chapter. Homography is a type of geometric constraint built between two views describing the relationships of all co-planar points. Many researchers drew on characteristic to build up visual SLAMs working under planar scenes. The camera pose and plane equation can be easily decomposed from the Homography constraint once the normal direction of the plane is given, *eg.* the ground plane. RANSAC (RANdom SAmple Consensus)-like robust estimation methods are also available for improving the robustness of the estimation against noisy input by categorising stochastically the inlier (noise-free) and outlier (noise-corrupted) data.

Classic homography works did not make their debut in the SLAM topic but first in the

field of Structure-from-Motion (SfM) [142], visual servoing [109] and even camera calibration [140]. Some homography-based visual odometry methods are introduced afterwards; most of them concentrate on the problem of ground plane driven navigation for vehicles or flying drones [104, 16]. In 2011, Christian Pirchheim and Gerhard Reitmayr present a mobile phone version of homography-based monocular SLAM and AR system [86]. It is equipped with a BA system and augmented reality module for displaying a virtual character on a planar map. The homography is estimated via FAST features and the RANSAC method for rejecting outliers. Dense mapping is gained by reprojecting all pixels information through the homography constraint.

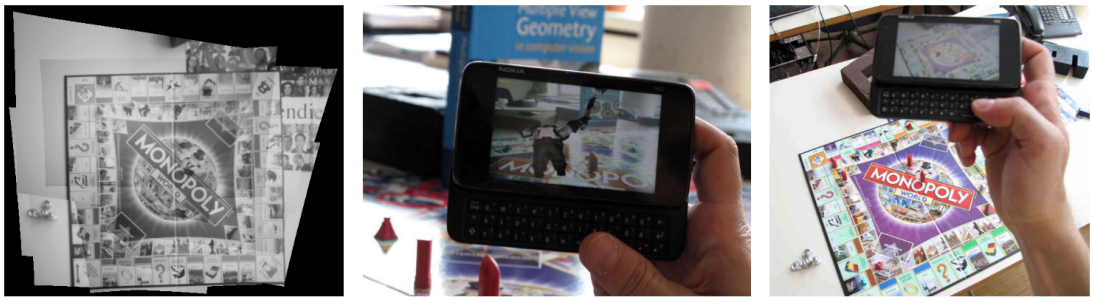


Figure 4.5 – A demonstration of a homography-based SLAM [86] and AR system implemented in a mobile phone. It recovers camera poses, then displays virtual character and does the dense mapping.

4.1.3 Loop Closure and Relocalization

Loop closure, the PTAM paper first mentioned in their feature-based SLAM system plays an important role in nowadays SLAM structure. The motivation behind is *drifting* during the tracking. Drifting describes the fact that all the errors are integrated and accumulated in a sequential estimation problem. Limited by the observing field (often Field-of-View of the camera), the moving nature of the agent robot (the scene can be occluded while moving) and the data association cost, most of the pose estimation problems are essentially based on iterative means to compute relative poses and estimate long trajectories. Therefore, once errors appear during the sequence, along the temporal direction, a *drift* seems inevitable and needs to be counterbalanced in odometry systems.

The loop closure technique solves this predicament. The main idea consists in retrieving already passed locations (often represented as keyframes) and taking them into account during the BA procedure for eliminating accumulated errors. Implicitly, this op-

eration can also robustify SLAM systems under non-static scenes by adding landmarks from different conditions and times to group a more informative map.

The requirements of loop closure module is very high in terms of precision and robustness, as the misleading results may trick the system and include wrong candidates into the optimization framework. The loop closure problem has multiple levels of difficulties influencing the precision and robustness for retrieving correct candidates: i) the quantity of passed keyframes may scale up to thousands of images w.r.t the limited storage of a SLAM system and increasing searching time of candidates; ii) many of the keyframes are taken from different views and distances; iii) perturbations by perception aliasing: it is triggered by repetitive textures often existing in artificial environments, such as the stacked bricks or windows on the facade of buildings.

To avoid storing all images directly for retrieval and achieving precision under noises, mainstream loop closures rely on the appearance-driven image retrieval method with the help of compressed information of the whole image or low-level handcrafted features for representing local regions.

For accelerating the matching performance, instead of using direct matching scheme which is inefficient and proportionally growing with the quantity of the visual features (for each keyframe, the feature number can go up to thousands), bag-of-words is one solution for this problem: The bag-of-words model converts a feature space, *eg.* low-level descriptors like ORB or SIFT, to a finite dimension of words. And instead of comparing all descriptors in a brute-force fashion, the features captured from each image are assigned to particular words (usually pre-offered by a set of words named dictionary) and create a set (bag) of assigned words. Similar to a frequential statistics measuring, this procedure sublates the geometric and spatial information among visual features. Therefore, one can handle the matching step as a histogram similarity or binary sequence matching (*eg.* Hamming distance) with bag-of-words. Perception Aliasing problem can be addressed in the bag-of-words too, by proposing the concept of TF-IDF (Term Frequency-Inverse Document Frequency) or other similar statistical tools such as Bayensien estimation. TF-IDF is a numerical statistic measure reflecting the importance of words to a document (in our case, a keyframe). In other words, TF-IDF gives less importance to the words occuring more frequently, which in our scenario are the perception aliasings.

FAB-MAP [26] is proposed in 2008 by Mark Cummins and Paul Newman for addressing the loop closure problem under the context of SLAM relocalization via the bag-of-words model. It calculates observational likelihood with implemented SIFT or SURF

features. A Chow-Liu tree [22] simplified joint probability distribution is proposed for handling the perception aliasing problem and learning a dictionary offline. For example, FAB-MAP considers the matching number of visual words and takes into account the fact if these words are rare enough to be observed for distinguishing two candidates. A likelihood probability is computed by using a normalising constant denominator for representing all visited locations. A threshold is available on this probability to decide if a revisited place is in sight. FAB-MAP shows good performance and high versatility on various visual features. Though FAB-MAP set a baseline for relocalization methods, some aspects can be improved if we see the proposed system retrospectively: i) Like all bag-of-words methods, ignoring spatial information helps on efficient matching and detecting, but ambiguities can exist when the robot agent runs in similar scenes for a long time. ii) speed performance with features: SIFT and SURF are good features with high matching performance but relative heavy to compute and process (hundreds of milliseconds for each image). A 3D versioned modification of FAB-MAP partially answers the first question; FAB-MAP 3D [83] incorporates the 3D information acquired by the lidar sensor and converts it into distance distribution. The second question is addressed by the work of the following paragraphs: DBoW [44].

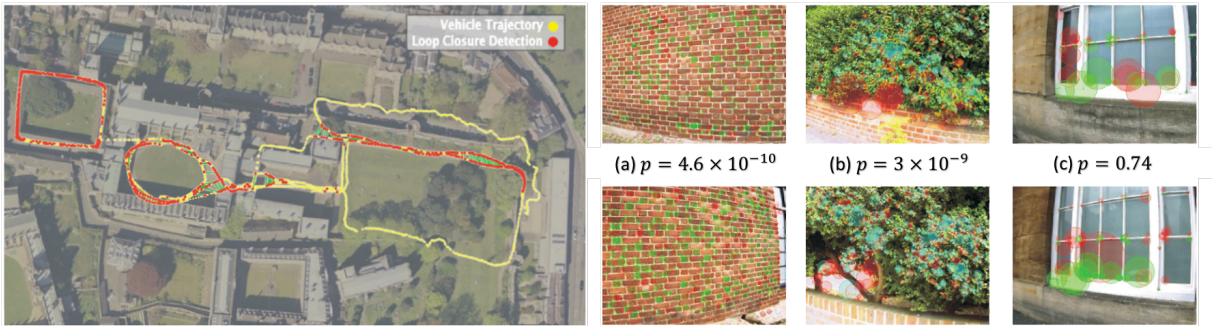


Figure 4.6 – The FAB-MAP relocalization results and demonstration of its robustness against perception aliasing effect: left side shows vehicle trajectory on yellow and found relocalization on red colour. On the right side, three pairs of candidates are given with their calculated probability by FAB-MAP. It clearly shows that even the appearance of two images roughly looks pretty similar to naked eyes, the FAB-MAP system can differentiate the correct candidates from the wrong ones (Figure from [26]).

The DBoW [44] is proposed jointly (though different papers) with the ORB-SLAM. DBoW focuses on ORB and BRIEF features compatible to the ORB-SLAM structure. Oppositely to the SURF and SIFT feature in the FAB-MAP, the most significant characteristic of the ORB feature is its high-speed performance without compromising repeata-

bility and rematching quality. Thanks to BRIEF-like descriptors' binary nature, DBoW proposes building a vocabulary tree for faster retrieval via a hierarchical matching scheme and clustering skills. Many other empirical rules are offered together with the paper for a complete loop closure structure: i) the idea of grouping frames together as a whole entity to process; ii) normalisation by the similarity score of the neighbour frame as a threshold control and rotational frame rejection; iii) temporal consistency check for not only applying on current candidate but also requiring to match short sequence around the target candidate. Direct and inverse index techniques are also used for accelerating the matching process and TF-IDF technique are applied for solving the perception aliasing. DBoW gained state-of-the-art relocalization performance in terms of speed and accuracy. It is widely used in ORB-SLAM and its derivatives in some following papers.

Many other loop closure-like methods are present in the recent literature: NetVLAD [5], RatSLAM [75], and a series of neural-network-driven image retrieval systems [115, 114] and 2D-3D relocalization problems [102, 64], etc.

4.2 Illumination Robustness

Image is a matrix collecting integrated and discretized photon beams through a given optic system. As it is generated by illumination, image information is naturally prone to be influenced by lighting too. This section will discuss illumination influence during the visionary robotics and SLAM concerning tasks, from brief mechanism to possible related works we are interested in.

4.2.1 Brightness Is Not Constant

James Kajiya, a pioneer researcher in computer graphics domain, developed a physic model describing luminance under a geometric optics approximation for computer graphics usage in 1986: a rendering equation [59] with BRDF (Bidirectional Reflection Distribution Function) [81].

The rendering equation is originated from the law of conservation of energy, it draws a formula of the leaving radiance from a point as the sum of emitted plus reflected radiance in an integral equation (see Fig. 4.7):

$$L_{\mathrm{o}}\left(\mathbf{x}, \omega_{\mathrm{o}}\right) = L_{\mathrm{e}}\left(\mathbf{x}, \omega_{\mathrm{o}}\right) + \int_{\Omega} f_{\mathrm{r}}\left(\mathbf{x}, \omega_{\mathrm{i}}, \omega_{\mathrm{o}}\right) L_{\mathrm{i}}\left(\mathbf{x}, \omega_{\mathrm{i}}\right) \left(\omega_{\mathrm{i}} \cdot \mathbf{n}\right) \mathrm{d}\omega_{\mathrm{i}} \tag{4.1}$$

where the L_o is the outgoing radiance along the direction ω_o of the location $\mathbf{x}$ (radiance seen by the observer), L_e is emitted radiance, and L_i is the inward radiance from direction ω_i, the very negative direction against the outward angle ω_o. The intergration is over a hemisphere region $\int_\Omega$, f_r is the bidirectional reflectance distribution function, the proportion of light reflected from inward to outward direction.

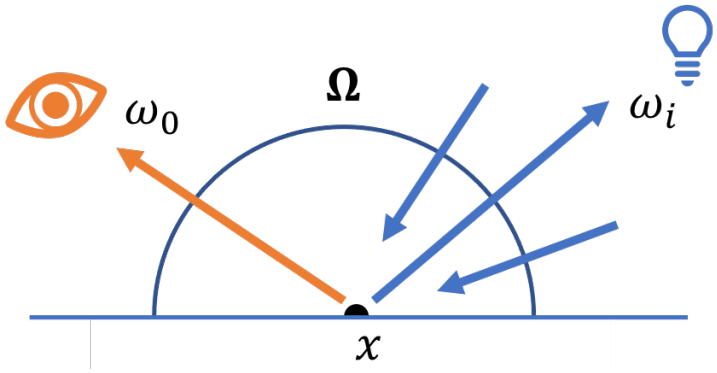


Figure 4.7 – Render equation depicting the observer's incoming radiance as an addition between emitted radiance at location $\mathbf{x}$, a hemisphere integration of BRDF function and incoming light.

Without going too deep into the detail, one can draw a quick conclusion from the render equation that the radiance level that a observer sees is controlled by multiple parameters: observing direction, global illumination level, materials parameters hidden in the BRDF function (diffuse or mirror reflection) etc. One should realise that the *Brightness Constancy Assumption* which is claimed largely in the stereo vision and SLAM field, is mercurial at a certain level. Let alone the seasonal and day-night nature lighting changes, illumination robustness does influence a considerable amount of robotic vision and SLAM tasks.

4.2.2 Illumination Variance in Digital Images

One may argue that as long as we carefully choose the working environment of the robotic agent and control the illumination parameters (not trying to 3D reconstruct a scene of moving lights where everything is reflective, a discotheque, for example), we can solve the problem of changeable lighting. However, other issues still occur at the stage of digitalization. The most frequent one is the compressed range of intensities: when images are taken under overly dark or bright areas, texture information risks to be damped or lost for that the gradient information is no longer obvious (See Fig 4.8 for an example).

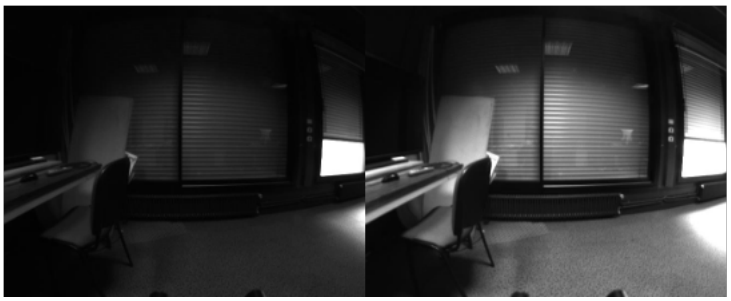


Figure 4.8 – A pair of example shows over dark and bright areas in two pictures of different aperture parameter, pay attention to overly bright region at window area and overly dark region at corner.

The main reason is that the dynmaic range of digital image from radiance energy is not only limited but also nonlinear. Reflecting on pictures, it means the radiance energy lower or higher than a certain level will be ignored or saturated. Moreover, different ranges may be compressed in different ways while the environmental illumination is changing.

For most feature-based SLAM methods, the importance lies in the illuminative influence towards visual features. Nevertheless, the designers of low-level features try to mitigate and adjust the situation by setting some thresholds at detecting and matching steps. It still can be seen that the illumination variance causes ill-functioning of the keypoint features. One straightforward example is given in Fig. 4.9, which shows the ORB feature detecting and matching between darker and brighter images yields some failures cases of the identical environment (yellow and red colour keypoints).

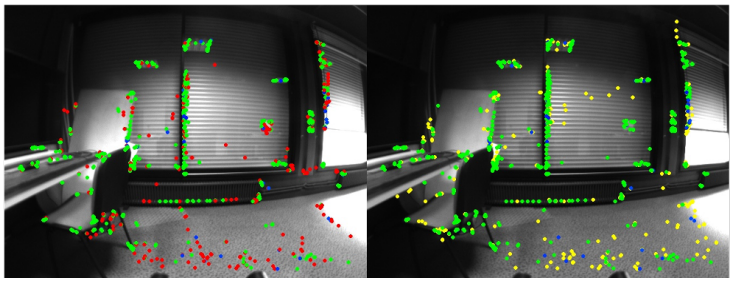


Figure 4.9 – An example of ORB feature on different illuminative condition, green dots mean correct tracked and matched features. Red and yellow represents untracked features detected at each image, respectively.

The issue has often been tackled at the extractor level by searching an optimal contrast threshold in the KP extractor w.r.t the current lighting condition. For example, in SuperFast [40] the FAST contrast threshold – a threshold value that triggers a brighter,

darker or similar decision on per-pixel comparison – is dynamically computed using a feedback-like optimization method that yields a new threshold value per region in the image. Lowering the threshold however tends to generate a large number of detections that influences the computational capacity of other processes.

Another possibility consists in applying image transformations (*eg.* contrast enhancers) on captured images before applying keypoint detectors. Interestingly, it has been demonstrated that keypoint extractors gain significant performance by using High Dynamic Range (HDR) images as input, converted to Standard Dynamic Range (SDR) images through tone-mapping operators [93, 87].

Among these techniques, a learning-based optimal tone-mapping operator has been proposed for SIFT-like detectors [94]. But the high computational cost and specific HDR devices required, as well as HDR-customized extractors hamper the wider applicability of such approaches. In comparison, for SDR images, research has mainly focused on contrast enhancement operators for aesthetic and perceptional goals through changes in the exposure times [138] which remain limited in addressing robustness problem in keypoints tracking of the SLAM context.

Direct SLAMs usually suffer more illumination changes since most direct SLAM methods utilise photometric error alignment for computing camera poses. Variant illumination can bring disastrous results if this process is not correctly executed. [14] proposed an online calibration method in particular for dense SLAM methods. The algorithm recovers the exposure times of consecutive frames, computes the camera response function (CRF), and the attenuation factors for handling the vignetting problem. An experiment of Kanade-Lucas-Tomasi (KLT) feature tracking result is given in Fig. 4.10 for showing its efficiency and ability to improve image aligning quality.

4.2.3 Photometric Error vs Mutual Information

As many blame the vulnerability of photometric error, which is also equivalent to SSD (Sum of Squared Differences) or SAD (Sum of Absolut Differences) in some scenarios, other genres of measure are proposed and applied in the visual tracking tasks, including: ZNCC (Zero Mean Normalized Cross-Correlation) and MI (Mutual Information).

A ZNCC measure between two images $\mathbf{I}$ and $\mathbf{I}^*$ is defined as:

$$ZNCC\left(\mathbf{I},\mathbf{I}^*\right) = \frac{\sum_{\mathbf{x}}(\mathbf{I}(\mathbf{x}) - \bar{\mathbf{I}})\left(\mathbf{I}^*(\mathbf{x}) - \overline{\mathbf{I}^*}\right)}{\sigma_{\mathbf{I}}\sigma_{\mathbf{I}^*}} \tag{4.2}$$

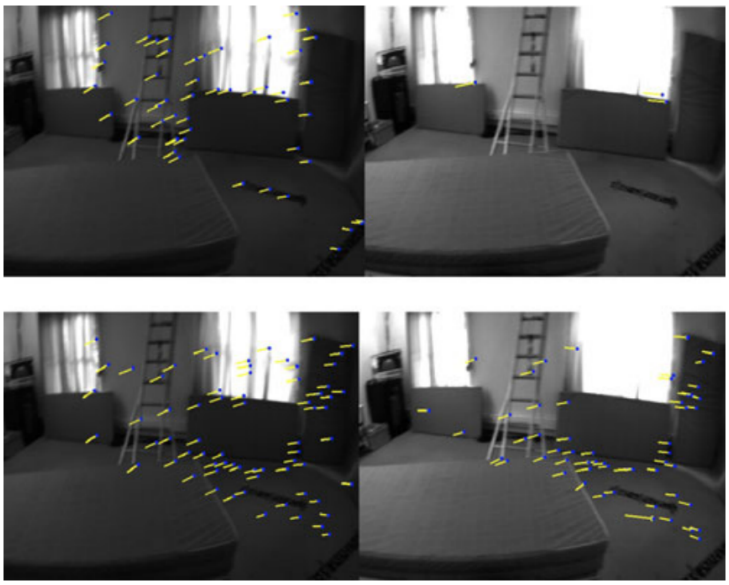


Figure 4.10 – Difference of KLT tracking results before and after gain adapting operation. First row: KLT tracking on original image; second row: gain adaptively transformed of identical images. Note that the traditional KLT method is prone to be influenced by intensity changes as it anchors on photometric error measure (Figure from [14]).

where $\bar{\mathbf{I}}$ and $\sigma_{\mathbf{I}}$ represent the mean and variance of the image $\mathbf{I}$, same for all of $\mathbf{I}^*$. An intuitive explaination of the ZNCC can be regarded as a variance-aware normalization measure of the original image. Naturally it ameliorates the robustness against global illumination variance.

Another measure is the MI (Mutual Information); under image scenario, it's defined as the addition of two image's entropy subtracting the joint entropy of all of two.

$$MI\left(I, I^*\right) = h(I) + h\left(I^*\right) - h\left(I, I^*\right) \tag{4.3}$$

where $h(I)$ and $h\left(I, I^*\right)$ are the entropy of one image and the joint entropy of two images, respectively. We will discuss with more details in the next chapter about their definition and calculation from image representation.

In the paper [29], a comparison among SSD (photometric error), ZNCC and MI measures of image registration is given between an aerial image and an abstract map reference (See Fig. 4.11)

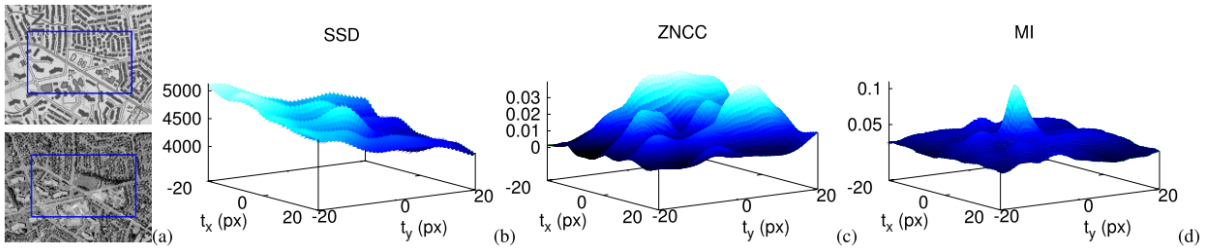


Figure 4.11 – Image registration results w.r.t translational changes between an aerial images and a map layout: MI shows a decent optimum near the zero translational distance, whereas the SSD and ZNCC give relative less apparent responses for optimization purpose. (Figure from [29]).

4.2.4 Loop Closure by Heterogeneous Data

Except for addressing via the image level, some loop closure and place recognition methods seek heterogeneous data representation for achieving more robust performance against not only lighting changes but also seasonal change and day-night shifts. One example is that we mentioned above, the FAB-MAP 3D. It relies on 3D lidar data for eliminating ambiguities caused by the lack of structural information.

Figure 4.12 – Examples of day-night shift and seasonal changes from RobotCar Seasons [70] (first row) and CMU Seasons datasets [102] (second row)

[47] introduce a graph method for achieving relocalization with semantic information. Semantic information can be regarded as a label map of semantic regions detected or synthesised from RGB images. They exploit random walk skill on a locally merged semantic region graph from 2D images for extracting label sequences as descriptors. A matching scheme of identical descriptor between two subgraphs is then utilised for retrieving possible candidates. Finally, geometric checks are also available in their structure for making

sure to find geometrically consensus results. They achieved good performance, especially under the conditions of illuminative, day-night and seasonal changes, as semantic information preserves high robustness against these noises. Liu et al. [67] also adopted a similar idea and added depth information and created a three-dimensional graph, instead of [47] build graph structure from a sequence of 2D images. But the requirements of RGB-D images implicitly constraints the usage environment to indoor scenes, as RGB-D cameras often fail to report far distance information.

4.3 Conclusion

This chapter discussed the development of visual SLAM techniques, categorization of SLAM methods, loop closure and relocalization, and influence of illumination variance. In the development of SLAM techniques, we give some milestone-level research works and draw a history timeline of this technique, including their evolutions, motivations, and paradigms. We then elaborate on the advantages with disadvantages of different type of SLAMs: feature-based and direct. The importance and rationale of the relocalization system are discussed, with some examples given and explained. Finally, we focus on the illumination variance and discuss its influence and possible solutions; hope to provide a general view of the topic we want to address via this thesis.

Chapter 5

Organisation of the thesis

In this chapter, we discuss the organisation of this thesis and give a roadmap for the following chapters. As illustrated in the background chapter, SLAM systems are complex and comprise many modules, therefore we propose to interpret the SLAM problem from different reciprocal angles (See Fig. 5.1):

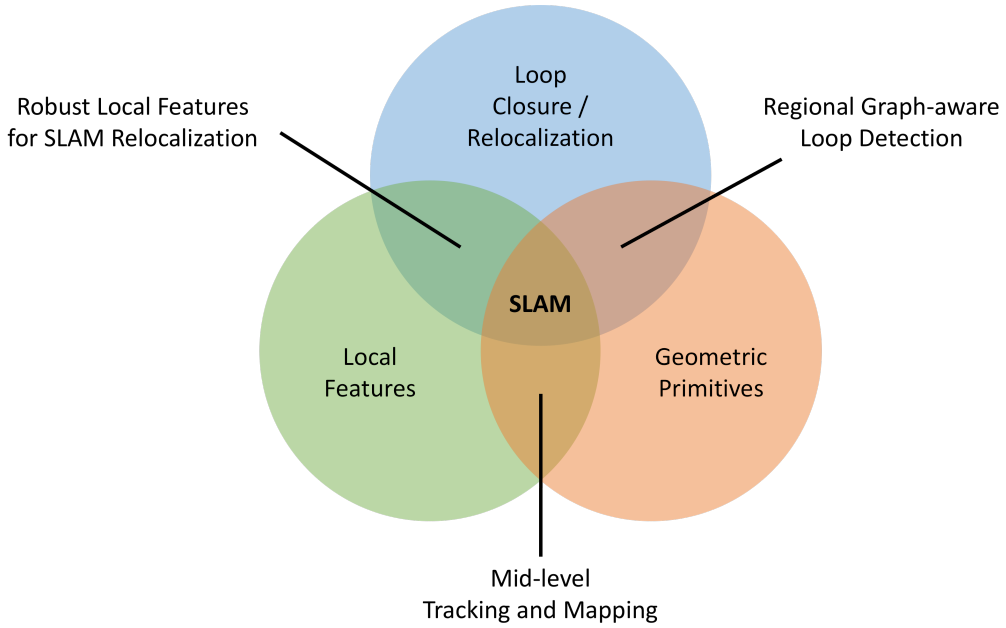


Figure 5.1 – We decompose SLAM technique into three aspects and present the intersected regions as our contributions: i) robust local feature for SLAM relocalization purpose; ii) mid-level geometric primitive for tracking and mapping more semantically; iii) regional graph-based binary descriptors for loop detection tasks.

Our main contributions can be categorized into three different but related modules in the SLAM system: local features for relocalization, mid-level features via keypoints and templates for tracking and mapping, and loop detection with a graph-aware binary descriptor.

Robust Local Feature for SLAM Relocalization

In the first category, we present some improvements on the low-level keypoint detection and matching under feature-based SLAM context (*eg.* ORB, SIFT etc.). We propose a multi-layered image representation (MLI) that computes and stores different contrast-enhanced versions of an original image in relevant chapters. Keypoint detection is performed on each layer, yielding better robustness to light changes. An optimization technique is also proposed to compute the best contrast enhancements to apply to each layer with naive optimization on matched keypoints and with mutual information as an approximation respectively.

Mid-level Geometric Primitive for Tracking and Mapping

In the second category, we exploit the multiple planar assumption reinforced by superpixel techniques to achieve a balance between feature-based sparse methods and dense mapping results. Via multiple homographies from two RGB images, we manage to handle the decomposition ambiguity, relative pose estimating and non-linear optimization process. An extension with template tracker and clustering techniques is introduced in the relevant chapters too.

Regional Graph-based Binary Descriptors for Loop Detection

Finally, we introduce a novel binary graph descriptor on segmented images and its implementation with incremental Bag-of-Words method. Experiments demonstrate that under dynamic environments, including lighting variations and season changes, the proposed descriptor and loop detection system is able to recover the passed locations with the help of multiple heterogeneous data formats: including color images, depth images, semantic segmentations and even deep neural network (DNN) descriptors.

Chapter 6

Multiple Layers of Contrasted Images for Robust Feature-based Visual Tracking

6.1 Problem Description

Research in visual tracking systems such as SLAM and SfM (Structure-from-Motion) has led to mature technologies exploited in industrial-level systems. Except for direct methods working on the analysis of changes in pixel gradients, the majority of visual SLAMs rely on corner detection with *extractors* that extract keypoints (KP) and *descriptors* that identify and match the extracted keypoints over different frames.

Unfortunately, the corner detection process and consequently the matching problem are strongly dependent on the illumination condition at the moment of capturing images and generally make a brightness constancy assumption. Although the matching process usually relies on gradient information that is more or less independent from intensity, SLAM and SfM methods still suffer from illumination changes at different degrees (see Fig. 6.1) and may yield inaccurate maps and even tracking failures during the tracking process [82, 106].

Robustness to light changing conditions is therefore a central issue that has received increased attention. The issue has often been tackled at the *extractor* level by searching an optimal contrast threshold in the keypoint extractor with respect to the current lighting condition. For example, in SuperFast [40] the FAST contrast threshold – a threshold value that triggers a brighter, darker or similar decision on per-pixel comparison – is dynamically computed using a feedback-like optimization method that yields a new threshold value per region in the image. Lowering the threshold however tends to generate a large number of keypoints that influence the computational capacity of other processes, and the proposed technique requires specific adaptations to be applied to other keypoint detectors.

(a) Daylight condition (b) Flashlight condition (c) ORB Matching (d) Our Method

Figure 6.1 – ORB keypoint extractor and descriptor on different lighting conditions (images a and b). Only a few keypoints are matched between (a) and (b) using ORB (*i.e.* same position, same descriptor), compared to our MLI method (image d).

Another possibility consists in applying image transformations (*eg.* contrast enhancers) on captured images before applying keypoint detectors. Interestingly, it has been demonstrated that keypoint extractors gain significant performance by using HDR images as input, converted to SDR images through tone-mapping operators [93]. Among these techniques, a learning-based optimal tone-mapping operator has been proposed for SIFT-like detectors [94]. But the high computational cost and specific HDR devices required, as well as HDR-customized extractors hamper the wider applicability of such approaches. In comparison, for SDR images, research has mainly focused on contrast enhancement operators for aesthetic and perceptional goals through changes in the exposure times [138] which remain limited in addressing robustness of keypoint tracking.

In the following sections, we seperate this chapter into two parts to present the concept of the proposed Multiple Layers of Contrasted Images, *i.e.* MLI, in Part I and a Mutual Information based computation method in Part II respectively.

6.2 Part I: Issues with Contrast Enhancers

Except for HDR images, the majority of contrast enhancement techniques can be defined as continuous monotone surjective mappings from domain interval $[0, 1]$ to codomain interval $[0, 1]$ that transform a given image to a (more) contrasted version. Typically, the classical S-Curve Tone Mapping method [138] is used to correct underexposure and overexposure regions in images, by applying a per-pixel function.

However, by using such transformations the improvement of the contrast in one region must necessarily be *paid* for by a reduction of the contrast in another region (see examples of S-Curves in Fig. 6.2). Given that most keypoint detection techniques are

based on analysis of finite local differences in contrasts, contrast enhancement tends to increase the detection of keypoints by passing some internal thresholds, while contrast compression leads to the opposite. We illustrate this through the example of a synthetic scene shot in different lighting conditions [84]. The ORB detector [100] is used to extract and match keypoints between a well-lit given reference image (see Fig. 6.1 a) and an image from the same viewpoint with only a flashlight illuminating the scene. The ORB detector with default parameters only finds a few *matched keypoints* between the two images (*i.e.* keypoints extracted and described as being the same at the same image locations). Interestingly by applying different S-Curve transformations (see Fig. 6.2), the total number of keypoint matches increases while ORB detector loses already matched keypoints from previous transformations. This empirically shows (i) that a single contrast enhancer only represents a partial solution to keypoint robustness, and that (ii) improving extraction by contrast-enhancement also improves matching.

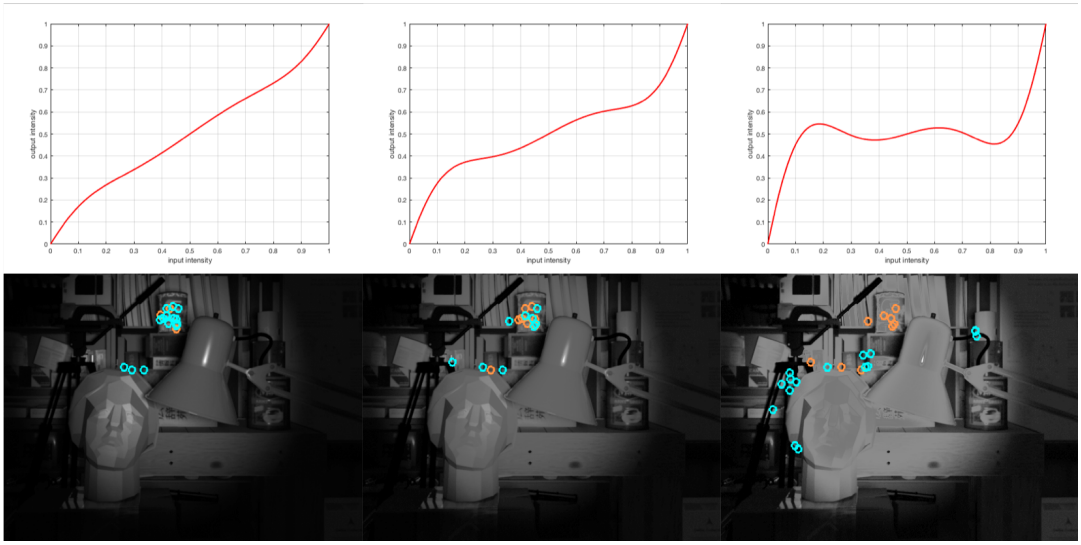


Figure 6.2 – Matching keypoints with ORB detector between a reference image (see Fig. 6.1 a) and different S-Curve tone mapped versions of an image in a different lighting condition. Each tone-mapping provides newly matched keypoints (blue) while losing others (orange).

6.3 Part I: Multi-Layered Image

In this chapter, we propose a Multi-Layered Image (referring to as MLI) to generate k contrast enhancements of a given image into k image layers on which keypoint detec-

tion will be performed. The contrast enhancement technique relies on a saturated affine brightness transfer per-pixel function (SAT). We use a SAT form that defines a *contrast*

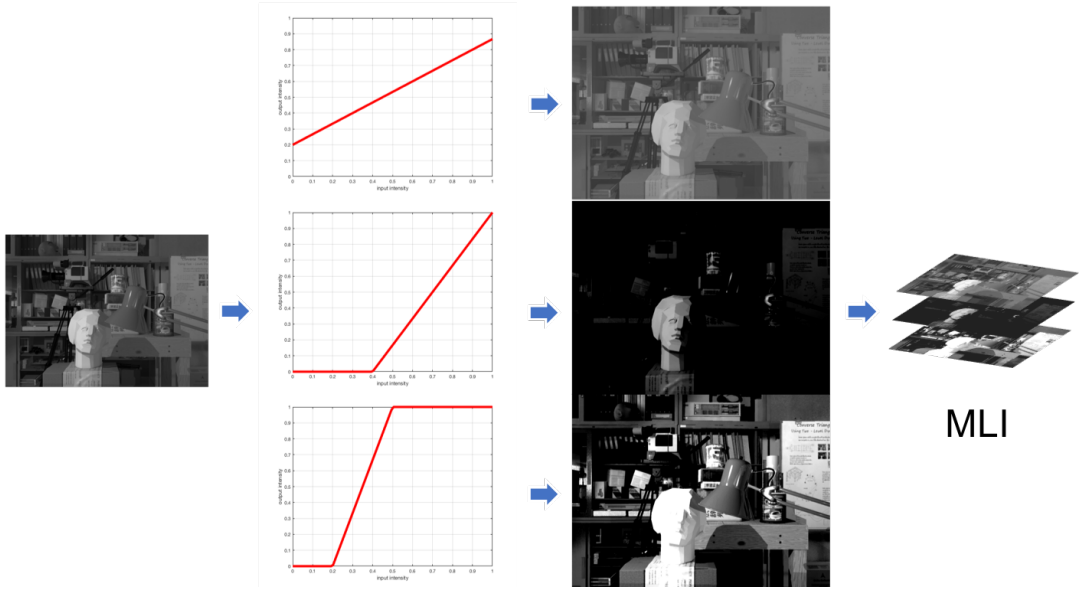


Figure 6.3 – Using SAT function with different contrast bands to generate a multi-layered image representation (MLI).

band $\mathbf{u} = (a, b)^\top$ which conveniently models the lower cut point (a) and higher cut point (b) of the saturation, with a linear interpolation between a and b on pixel intensity i (see Fig. 6.3). A given contrast $\mathbf{u} = (a, b)^\top$ is defined in a contrast space $\Gamma \subseteq \mathbb{R}^2$, where Γ is the space of all contrast bands where $b > a$.

$$f_{SAT}(i, (a, b)^\top) = min(max(0, i/(b - a)), 1) \tag{6.1}$$

Parameters a and b naturally represent the *band* region where the contrast is enhanced, which motivated the choice of this operator compared to S-Curve. To ensure enhancement or compression of contrasts, we define the range of values for $\mathbf{u} = (a, b)^\top$ as $a \in [-\infty, 1]$ and $b \in [0, \infty]$. The computation of a layer k in our MLI representation is performed by applying the following operator MLI_k on all pixel intensities of the image using a contrast band $\mathbf{u}_k$. A MLI is therefore represented as a set of k image layers where $MLI_k(I) = f_{SAT}(I, \mathbf{u}_k)$ for an image I, where $f_{SAT}(I, \mathbf{u}_k)$ is the application of f_{SAT} on all pixels of I.

6.4 Part I: Low-correlated Contrast Space

To address the issue of robustness, the key challenge is therefore to generate different layers such that each layer has the *lowest correlation* with the others in terms of detected keypoints (*i.e.* aiming at providing new keypoints in each layer). In other terms, we are looking at computing a set of contrast band parameters such that each contrast band yields an image containing newly matched keypoints with the reference image (the initial lighting condition).

We propose a technique to compute the optimal contrast bands together with a stopping criterion on the number of layers required, given a reference image I^* representing a given lighting condition and a camera image I in another lighting condition. The first layer is computed by selecting a contrast band $\mathbf{u}_i$ that maximizes the *correspondence* of keypoints between the reference image I^* and the contrast-enhanced image $f_{SAT}(I, \mathbf{u}_i)$. The other layers are computed by selecting contrast bands that provide the lowest correlation (in terms of correspondence between keypoints) with the current contrast band. More formally we start by defining the *keypoint-correspondence* set between two images. Given S^* the set of keypoints extracted from a reference image I^* (and respectively S from I), the *keypoint-correspondence* S_{Cor} is the set of keypoints in S^* for which there is a *correspondence* in S, *i.e.* for which there is a keypoint at a similar location in image I:

$$S_{Cor} = \{x^* \,|\, x^* \in S^*, x \in S, \|x^* - x\| < \epsilon\} \tag{6.2}$$

This definition can be used to express the *repeatability* ratio [105] between two sets of keypoints from two different images, a well-known metric in visual tracking [46]:

$$\frac{Card(\{x^* \in S^*, x \in S,\ s.t.\ \|x^* - x\| < \epsilon\})}{Card(S^*)} \tag{6.3}$$

More generally, given a keypoint extractor e, we can define a *band-correspondence set* $S^{\mathbf{u}}_{Cor}$ as the *keypoint-correspondence* set between a SAT contrast-enhanced version of I and a reference image I^* given $\mathbf{u}$:

$$S^{\mathbf{u}}_{Cor} = \{x^* \,|\, x^* \in e(I^*), x \in e(f_{SAT}(I, \mathbf{u})), \|x^* - x\| < \epsilon\} \tag{6.4}$$

Intuitively, this means the more keypoints there are in this *band correspondence set*, the

better is the contrast band $\mathbf{u}$ in yielding an image containing corresponding keypoints with a reference image. We therefore express the cardinality of this set $M_{Cor} : \Gamma \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ as $M_{Cor}(\mathbf{u}) = Card(S^{\mathbf{u}}_{Cor})$. The global maximum of this function represents the optimal contrast band $\mathbf{u}$ in terms of *keypoint-correspondence* and is used to compute the first layer of our MLI.

We then need a way to compute new contrast bands with low-correlation in the contrast space $\mathbf{u} \in \Gamma$. We define a covariance-like method on $S^{\mathbf{u}}_{Cor}$ that provides a *co-Correspondence* set $S^{\mathbf{u}_1,\mathbf{u}_2}_{coCor}$. This *co-Correspondence* computes the corresponding keypoints between two contrast bands $\mathbf{u}_1$ and $\mathbf{u}_2$ and a reference image I^*. We similarly define its cardinality $M_{coCor} : \Gamma \times \Gamma \subseteq \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$.

$$S^{\mathbf{u}_1,\mathbf{u}_2}_{coCor} = \{x_1 \in S^{\mathbf{u}_1}_{Cor} \, | x_2 \in S^{\mathbf{u}_2}_{Cor}, \|x_1 - x_2\| < \epsilon\} \tag{6.5}$$

$$M_{coCor}(\mathbf{u}_1, \mathbf{u}_2) = Card(S^{\mathbf{u}_1,\mathbf{u}_2}_{coCor}) \tag{6.6}$$

Algorithm 1 Optimal MLI

$i \leftarrow 0$; $C^0(\mathbf{u}) \leftarrow M_{Cor}(\mathbf{u})$;
while $i = 0$ **or** $C^i(\mathbf{u}_i) > k * C^{i-1}(\mathbf{u}_{i-1})$ **do**
  $\mathbf{u}_i \leftarrow \text{argmax}_{\mathbf{u}}(C^i(\mathbf{u}))$
  $C^{i+1}(\mathbf{u}) \leftarrow C^i(\mathbf{u}) - M^{\mathbf{u}_i}_{sim}(\mathbf{u})$
  $i \leftarrow i + 1$
end while
return $\{\mathbf{u}_k\}_{k=1..N}$

Using this co-correspondence definition we can compute how much a given contrast-band $\mathbf{u}_r$ yields information similar to all the other contrast bands, *i.e.* the number of keypoints generated by this contrast-band also found in others. This similarity measure $M^{\mathbf{u}_r}_{sim} : \Gamma \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ is expressed as:

$$M^{\mathbf{u}_r}_{sim}(\mathbf{u}) = Card(S^{\mathbf{u}_r,\mathbf{u}}_{coCor}) \tag{6.7}$$

A low similarity represents a low correlation between the contrast bands. Using these definitions, the computation of the different contrast bands consists in applying a sequence of two stage operations (see Alg. 1). The first stage selects an optimal contrast band $\mathbf{u}$ that maximizes a cost function $C(\mathbf{u})$, *i.e.* maximizes the correspondences between reference

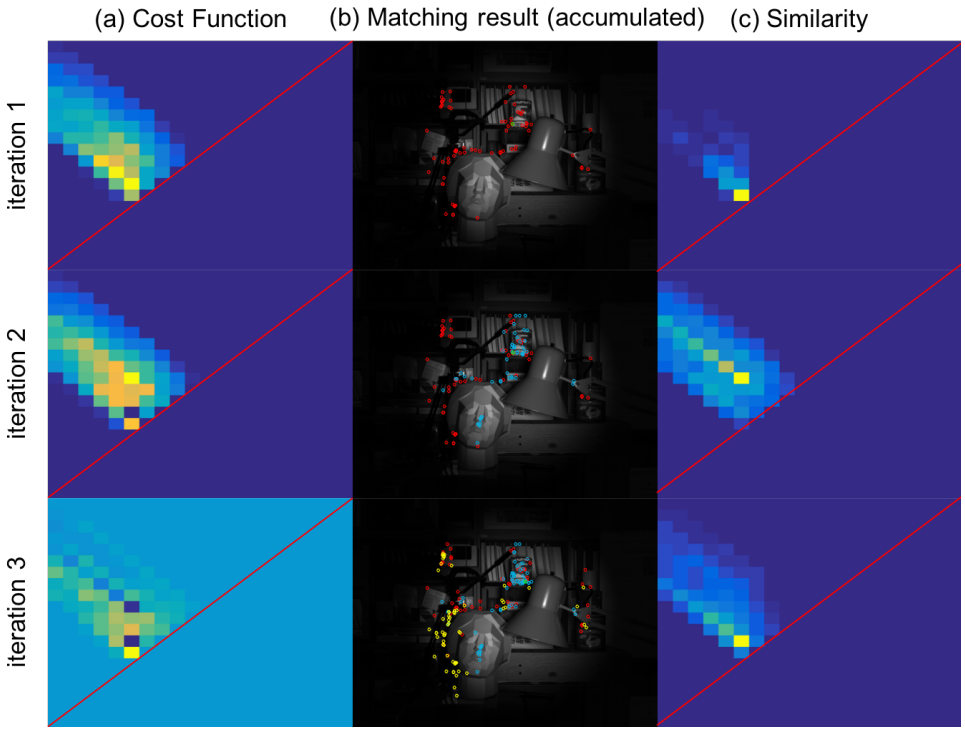


Figure 6.4 – Evolution of MLI layers: (a)(c) represent heatmaps of cost function $C^i(\mathbf{u})$ and similarity $M_{sim}^{\mathbf{u}_i}(\mathbf{u})$ in each iteration. (b) demonstrates accumulated matched ORB keypoints against reference image. In each heatmap, vertical and horizontal axis represents $\mathbf{u} = (a, b)^\top$ respectively with $a < b$.

image I^* and $f_{SAT}(I, \mathbf{u})$. The second stage then updates the cost function by subtracting the similarity $M_{sim}^{\mathbf{u}_i}$ between the current contrast band $\mathbf{u}_i$ and all others (ensuring a low correlation). The algorithm terminates when the new iteration yields information proportionally lower than the previous one using a factor k.

The algorithm is illustrated in Fig. 6.4 using the FAST extractor [98], and the ground truth is ensured by calculating BRIEF descriptor [17]. Again we reuse the data set of New Tsukuba [84]. For the purpose of illustration, Γ is defined as a discrete sampling space between $[-0.5, 1.5] \times [-0.5, 1.5]$. Images (a) and (c) represent the landscape of the cost functions $C^i(\mathbf{u})$ and similarities $M_{sim}^{\mathbf{u}_i}(\mathbf{u})$ of the previous optimal contrast band in each iteration. We observe that the maximums of $C^i(\mathbf{u})$ change every iteration after updating by a subtraction with M_{sim}. As one can see in the third iteration (Fig. 6.4.b), extract more keypoints with low-correlation between the layers. This empirically shows

cam \ ref		Daylight			Fluorescent			Lamps			Flashlight		
		ORB	SIFT	SURF	ORB	SIFT	SURF	ORB	SIFT	SURF	ORB	SIFT	SURF
Daylight	D	100/100	100/100	100/100	63.6/21.2	36.2/21.5	50.1/20.0	21.1/0.8	22.2/0.5	28.6/1.1	52.3/5.8	43.0/11.3	48.6/5.9
	M	100/100	100/100	100/100	**85.3/38.7**	**64.1/37.3**	**75.4/34.4**	**35.7/1.6**	**39.8/1.1**	**49.8/2.2**	**67.6/8.4**	**50.9/13.8**	**56.7/8.4**
Fluorescent	D	63.7/21.2	48.7/27.2	63.2/22.8	100/100	100/100	100/100	7.0/0.3	33.3/1.0	44.4/1.5	49.8/9.1	54.5/16.9	59.7/8.4
	M	**72.3/34.7**	**61.2/34.7**	**76.6/30.7**	100/100	100/100	100/100	**13.8/0.4**	**51.0/1.6**	**65.7/2.0**	**66.2/13.9**	**60.8/21.5**	**65.4/13.4**
Lamps	D	4.2/0.9	2.0/1.2	3.1/1.7	1.4/0.5	2.1/1.4	3.6/1.7	100/100	100/100	100/100	4.4/1.0	1.2/0.5	3.8/0.8
	M	**64.6/20.0**	**46.9/24.0**	**62.1/21.8**	**66.6/21.8**	**47.0/26.7**	**60.7/25.9**	100/100	100/100	100/100	**56.8/6.7**	**41.3/14.2**	**46.2/9.0**
Flashlight	D	34.0/5.3	12.2/5.4	16.4/5.3	32.4/7.6	11.3/6.3	16.2/6.2	14.6/0.5	5.1/0.0	12.1/0.2	100/100	100/100	100/100
	M	**58.1/11.8**	**31.4/11.8**	**44.6/11.4**	**61.4/16.6**	**30.5/15.0**	**44.0/13.5**	**18.4**/0.5	**16.4/0.5**	**35.0/0.8**	100/100	100/100	100/100

Table 6.1 – Repeatability/matching ratio evaluation between MLI (M) and default single image (D) in percentage.

that instead of intuitively or programmatically decreasing the thresholds parameters of detectors, an optimization scheme to compute the optimal contrast bands in each layer improves the correspondence of keypoints with a reference image.

6.5 Part I: Evaluations and Experiments

We first compare the use of MLI with classical detectors/descriptors: ORB [100], SIFT [68] and SURF [12] on the New Tsukuba Data set [84]. The process consists in measuring the *repeatability* ratio (*cf.* eq. 6.3) as well as the *matching* ratio by matching descriptors on feature points between a reference image I^* and new images I from the same viewpoint and different lighting conditions. For each condition, the optimal values of the contrast bands are computed by using the algorithm defined in Alg 1. The measures reported in Table 6.1 show that MLI improves repeatability and matching ratio across all detectors, despite different detection methods and default threshold parameters. This demonstrates the wide applicability of our approach.

We then compare the use of MLI on visual SLAM tasks in different lighting conditions. We choose ORB-SLAM [77] in which we implemented our MLI representation. We tested two sequential videos from a combination of four different lighting conditions. The experiment consisted in localizing and tracking the camera from the second video sequence against the keyframes generated from the first video sequence (in a way similar to NID-SLAM [82]). The measured value is the success rate, *i.e.* the percentage of the frames from second video successfully tracked against keyframes created from the first video.

The optimal contrast bands of the MLIs of each illumination condition (first video to second video) are computed by Algo. 1 over 5 sample images in the test set. Comparison

is performed between standard ORB-SLAM [77], our MLI implemented ORB-SLAM and reported results of monocular visual SLAM NID-SLAM [82] which demonstrated a good performance against illumination changing environments. Two to three layers are used in the experiments. The MLI approach is significantly more robust than the default ORB implementation (see Table 6.2), especially in difficult situations (Lamps to Flashlights, Daylight to Flashlight). Our approach also compares very favourably with NID-SLAM, displaying similar or better performances for a lower computational cost (keypoint extraction only represents 3 to 5% of computation time in ORB [77], limiting the impact of MLI cost). Typically the Lamps to Flashlight failed to track all keyframes with both ORB and NID (0%) and successfully tracked 94.2% of the keyframes with our MLI approach.

V_2 \ V_1	Daylight			Fluo			Lamps			Flash		
	NID	ORB	MLI	NID	ORB	MLI	NID	ORB	MLI	NID	ORB	MLI
Daylight	99.3	100	**100**	96.7	96.2	**98.4**	73.9	**97.6**	53.6	74.6	**79.8**	77.1
Fluo	95.0	88.1	**95.1**	99.7	100	**100**	85.3	93.9	**100**	95.8	100	**100**
Lamps	88.3	55.7	**93.3**	**93.6**	79.8	93.4	93.1	100	**100**	84.3	37.9	**96.8**
Flash	23.8	30.7	**77.6**	92.2	90.6	**93.6**	0.00	0.00	**94.2**	92.0	**100**	99.3

Table 6.2 – SLAM keyframe retrieval success rate among default ORB-SLAM, NID-SLAM and our MLI implementation.

6.6 Part II: Multi-Layer Images via Mutual Information

As we already prove the effectiveness of the concept of multi-layered image representation (MLI) in Part I, in the Part II we aim to accelerate the optimization process via the measure of Mutual Information (MI) instead of iterative searching of conditional maximum of matching results in previous part. We rely on the information theory approach of Mutual Information to compute the optimal contrast enhancements on each layer of the MLI. The specific contributions of this part are:

— an efficient process to compute optimal parameters for these contrast enhancements using an information theoretic approach;

— a dramatic improvement of ORB-SLAM tracking in light changing conditions;

Results displayed in Fig. 6.5 shows that our Mutual Information assisted MLI outperforms the default ORB-SLAM technique in terms of a stronger robustness to light

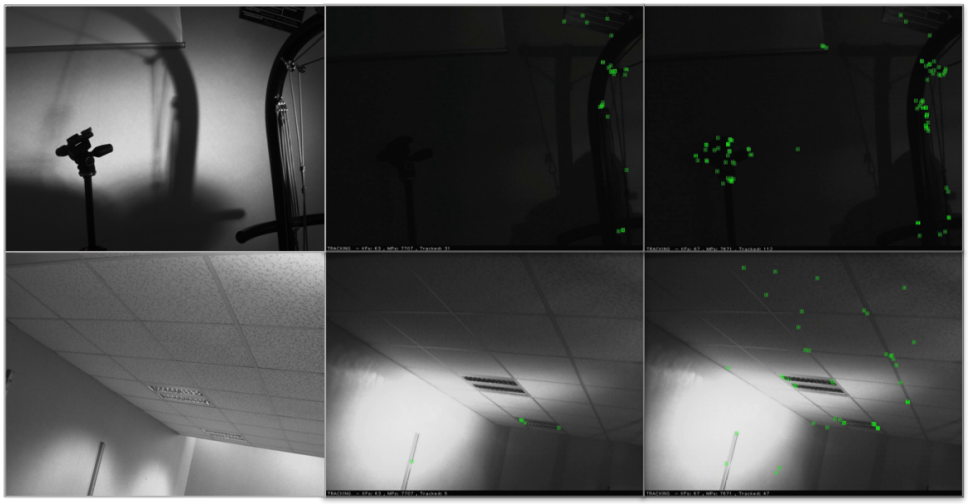


Figure 6.5 – Keypoint tracking results in different lighting conditions. Only a few keypoints are matched between reference condition (image a) and standard ORB-SLAM (image b), compared to our novel MLI method (image c).

changing conditions.

Adopted from the Part I, we recall that the concept of our Multi-Layer Images (MLI) consists in computing, for every frame, a number of contrast enhancements of the original camera image into different layers before applying keypoint detection on each layer. Different to previous Part I, sampling the keypoint matching results in a straightforward way, in this part we rely on a Mutual Information (MI) metric. The MI is an information theory measure of dependence between two random variables (images in our case), and demonstrate the relevance of this metric in keypoint tracking (see Section 6.7). The parameters of the next layer are then searched by maximising the mutual information with the reference image, without the information already provided by the first one. Other layers are computed in a similar incremental way. Given that landscapes of the Mutual Information metrics are difficult to optimize, a specific smoothing process is proposed that enables the use of straightforward gradient descent optimization techniques

6.7 Part II: Optimal Image Enhancement

The challenge consists in computing the best parameters for each contrast enhancement on each camera image in a way to improve detection and matching of keypoints.

Our hypothesis is that we can compute a close to optimal value of $\mathbf{u}_k$ by maximising Mutual Information between a well lit reference image I^* and a transformed test image $f_{SAT}(I, \mathbf{u}_k)$.

6.7.1 Mutual Information

Mutual information (MI) was initially introduced in information theory [107], and then widely applied in the field of computer vision for image alignment, model registration as well as visual tracking and SLAM [82, 29, 117]. The MI built from the image entropy of two different images provides a measure of their mutual dependence. In image alignment tasks, the higher the mutual information, the better the alignment since mutual information considers the distribution of the intensities as well as the intensities themselves.

Entropy $h(I)$ is a variability measure of a random variable I. In image alignment or illumination evaluation scenarios, I is regarded as one image with r the possible values (gray-level intensities) of I. Equation $p_I(r)=P(I=r)$ therefore expresses the probability distribution function of r, in other words the normalized histogram of the image. The Shannon entropy $h(I)$ of an image I is expressed as:

$$h(I) = -\sum_r p_I(r) log(p_I(r)) \tag{6.8}$$

With the same principle, the joint entropy $h(I, I^*)$ of two images I and I^* can be defined in the following way:

$$h(I, I^*) = -\sum_{t,r} p_{II^*}(t, r) log(p_{II^*}(t, r)) \tag{6.9}$$

where t and r are the possible grey-level intensities of the I and I^*. The joint probability distribution function is defined as $p_{II^*}(t, r)=P(I=t \cap I^*=r)$, which can also be regarded as a normalized bi-dimensional histogram of images I and I^*.

With the above notations of entropy and joint entropy, the mutual information (MI) is expressed as the intersection of two random variables I and I^* (see Fig. 6.6):

$$MI(I, I^*) = h(I) + h(I^*) - h(I, I^*) \tag{6.10}$$

6.7.2 Optimal Enhancement for the First Layer

We need to search for the optimal parameter $\mathbf{u}^*$ that maximises the MI between a reference image I^* (*eg.* an image lit in normal lighting conditions) and a contrast enhanced version of camera image $f_{SAT}(I, \mathbf{u})$.

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmax}}\, MI(f_{SAT}(I, \mathbf{u}), I^*) \tag{6.11}$$

We can empirically show that the MI has similar behavior to the ground truth wrt illumination changes. Given a reference image I^* under a given light condition, and a test image I in a different lighting condition, the computation of the ground truth (*i.e.* the absolute optimal enhancement) can be performed by an exhaustive sampling of the contrast band parameter $\mathbf{u}_k$, applying corresponding image transforms on I and evaluating the number of matched keypoints between I^* and $f_{SAT}(I, \mathbf{u}_k)$, as displayed in Fig. 6.10.

We illustrate this on an example from the NewTsukuba data set [84]. We compare the landscapes generated by sampling $\mathbf{u}_k$ on (1) the ground truth ORB detector and on (2) mutual information Eq. (6.10). Despite differences, we observe the optimums are positioned at similar contrast band values. In an obvious way, the more information is shared between a reference image and a contrast-enhanced image, the better are the detection and matching.

6.7.3 Optimal Enhancements for other Layers

The parameters of the second layer are searched by maximising the Mutual Information with the reference image, as well as the information already provided by the first layer (see Fig. 6.6).

This can be expressed as multivariate mutual information with the definition of higher dimensional joint probability distribution, to account for multiple image layers. From Eq. (6.8) and (6.9), a joint entropy of 3 random image variables is obtained with a definition of normalized tri-dimensional histogram $p_{II^*I^0}(t, r, w)$=$P(I$=$t \cap I^*$=$r \cap I^0$=$w)$ where t, r, w are possible gray-levels of each image respectively.

$$h(I, I^*, I^0) = -\sum_{t,r,w} p_{II^*I^0}(t, r, w) log(p_{II^*I^0}(t, r, w)) \tag{6.12}$$

Similarly, a multivariate mutual information between three images can be formulated:

$$\begin{aligned} MI(I, I^*, I^0) = h(I, I^*, I^0) + h(I) + h(I^*) + h(I^0) \\ -h(I, I^*) - h(I, I^0) - h(I^0, I^*) \end{aligned} \tag{6.13}$$

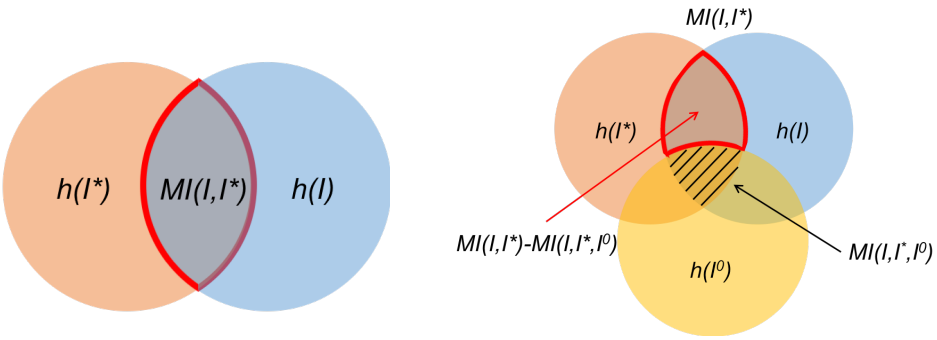


Figure 6.6 – Mutual information between two images (left), and three images (right) defined as the shared entropy between images.

Given I^* an image under reference light conditions, I the test image for which the contrast bands need to be computed, and $I^0 = f_{SAT}(I, \mathbf{u}_0)$ the first contrast band computed by Eq. (6.15), we can express the tri-variable mutual information to represent the low-correlated information generated in second layer (see Fig. 6.6 right red line).

$$\begin{aligned} MI(I, I^*) - MI(I, I^*, I^0) = MI(I^*, I|I^0) \\ = h(I, I^0) + h(I^*, I^0) - h(I, I^*, I^0) - h(I^0) \end{aligned} \tag{6.14}$$

Given the contrast band from first layer $\mathbf{u}_0$, the optimization of second layer is carried out as follows:

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmax}}\, MI(I^*, f_{SAT}(I, \mathbf{u})|f_{SAT}(I, \mathbf{u}^0)) \tag{6.15}$$

The computation of further layers can be expressed in a similar way. The mutual information between four images $MI(I, I^*, I^0, I^1)$ or more is computationally expensive to achieve. However a reasonable approximation can be computed using mutual information of previous optimal image $MI(I, I^*, I^1)$ to replace $MI(I, I^*, I^0, I^1)$, which balances the computational cost and preciseness.

6.7.4 Smoothing Mutual Information

Figure 6.7 – Images I^* and I from NewTsukuba data set [84]: a synthetic data set with identical camera trajectories and variant illumination conditions

Derivative optimization approaches favor smoother objective function landscapes to ensure an efficient descent to the optimum. Unlike image alignment where reducing the number of bins is important to smooth the cost function (image alignment indeed concentrates more on the geometric information instead of illumination information in one image [57, 29, 82]), for illumination estimation, lowering the histogram bins during the estimation does not provide any benefits and loses information. This is illustrated on another example from the NewTsukuba data set in Fig. 6.7 by presenting the landscapes of the cost function $MI(I^*, f_{SAT}(I, \mathbf{u}))$ (see Fig. 6.8).

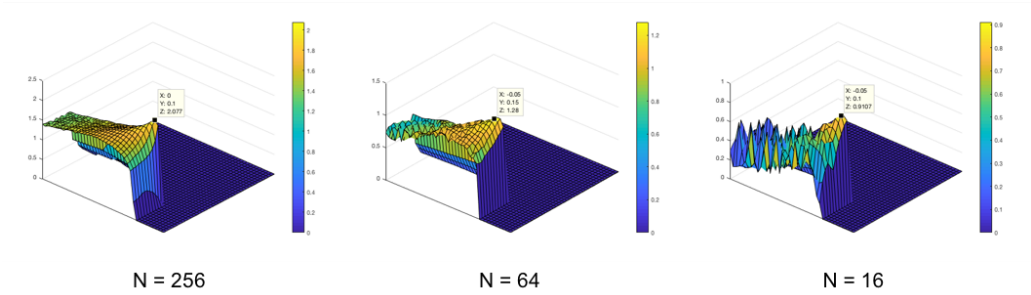


Figure 6.8 – Lowering the number of histogram bins (which is classical for image alignment tasks) leads to a difficult-to-optimize cost function landscape during illumination estimation process (axis represent parameters a and b of contrast band $\mathbf{u} = (a, b)^\top$).

In our work, we selected 256 bins for the purpose of illumination estimation combined with an *sigmoid-smoothed* SAT function Eq. (6.17). $f_{SgSAT}(I, \mathbf{u})$ based on Eq. (6.1) supporting a more curved transition at cutting points which leads to a smoother and less

aliased cost function landscape (see Fig. 6.9). Here we show our sigmoid function $Sg(x)$ with $k = 1/(b-a)$:

$$Sg(x) = \frac{1}{1 + 8ke^{(-(x-(b+a)/2)))}} \quad with\ \mathbf{u} = (a,b)^\top \tag{6.16}$$

With the definition of d as the length of contrast band, $d = b - a$, we have the *sigmoid-smoothed* SAT function $f_{SgSAT}(I, \mathbf{u})$ defined as:

$$\begin{aligned} f_{SgSAT}(I,\mathbf{u}) = &\ max(1-d,0) \times Sg(I) \\ &+ min(d,1) \times f_{SAT}(I,\mathbf{u}) \end{aligned} \tag{6.17}$$

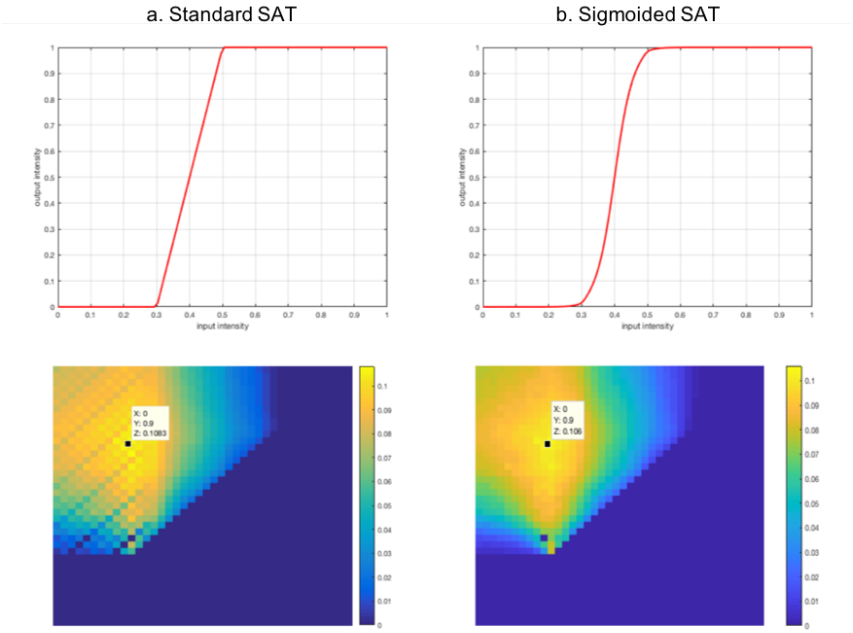


Figure 6.9 – Comparison between standard SAT function and sigmoid-smoothed SAT function wrt $\mathbf{u} = (0.3, 0.5)^\top$. Second row shows the conditional mutual information $MI(I^*, I|I^0)$ computed by Eq. (6.14), notations keep as aforecited for I^* and I with I^0 generated by the optimal contrast band from $MI(I^*, f_{SAT}(I, \mathbf{u}))$. We see clearly that standard SAT causes aliasing-like effect in the landscape, due to derivative discontinuity at cutting points a, b

6.7.5 Optimization Framework

Using the concepts introduced, we propose an optimization framework that computes a multi-layered image representation for each frame of a video sequence.

As illustrated in Algo. 1, the first step relies on the cost function of standard mutual information Eq. (6.11) to compute the optimal result of the first layer. The following layers are then computed by optimizing a cost function measuring conditional mutual information with the previous result. This aims at finding the best contrast band (*i.e.* with the lowest information correlation to the previously computed contrast bands). I^* represents the image under reference light condition and I the current test image to optimize, $\mathbf{u}_i, i = 0..N$ is referring to the computed optimal contrast band of each layer with a layer number N.

Algorithm 2 Optimal MLI Generated by MI

$i \leftarrow 0$
$\mathbf{u}_0 \leftarrow \text{argmax}_{\mathbf{u}}(MI(I^*, f_{SgSAT}(I, \mathbf{u})))$
while $i < N$ **do**
  $\mathbf{u}_i \leftarrow$
  $\text{argmax}_{\mathbf{u}}(MI(I^*, f_{SgSAT}(I, \mathbf{u})|f_{SgSAT}(I, \mathbf{u}_{i-1})))$
  $i \leftarrow i + 1$
end while
return $\{\mathbf{u}_k\}_{k=1..N}$

A demonstration with the NewTsukuba data set (see Fig. 6.7) in Fig. 6.10 illustrates the idea of our multiple step optimization framework. First step is the computation of MI between two images, shown in Fig. 6.10 representing the first layer (layer 1). The second and third layer rely on the computation of the first layer and instead of optimizing a standard MI cost function, a conditional mutual information cost function is optimized using Eq. (6.15). Comparing with the ground truth generated by ORB [100] detector, our proposed method presents a highly similar behavior as well as a characteristic of derivability for gradient optimization methods.

6.8 Part II: Evaluation and Experiments

To evaluate the benefits of our approach in visual SLAM relocalization tasks, like the Part I, we first select a synthetic scene benchmark under different static and dynamic lighting conditions [84]. This dataset encompasses four videos rendered with identical

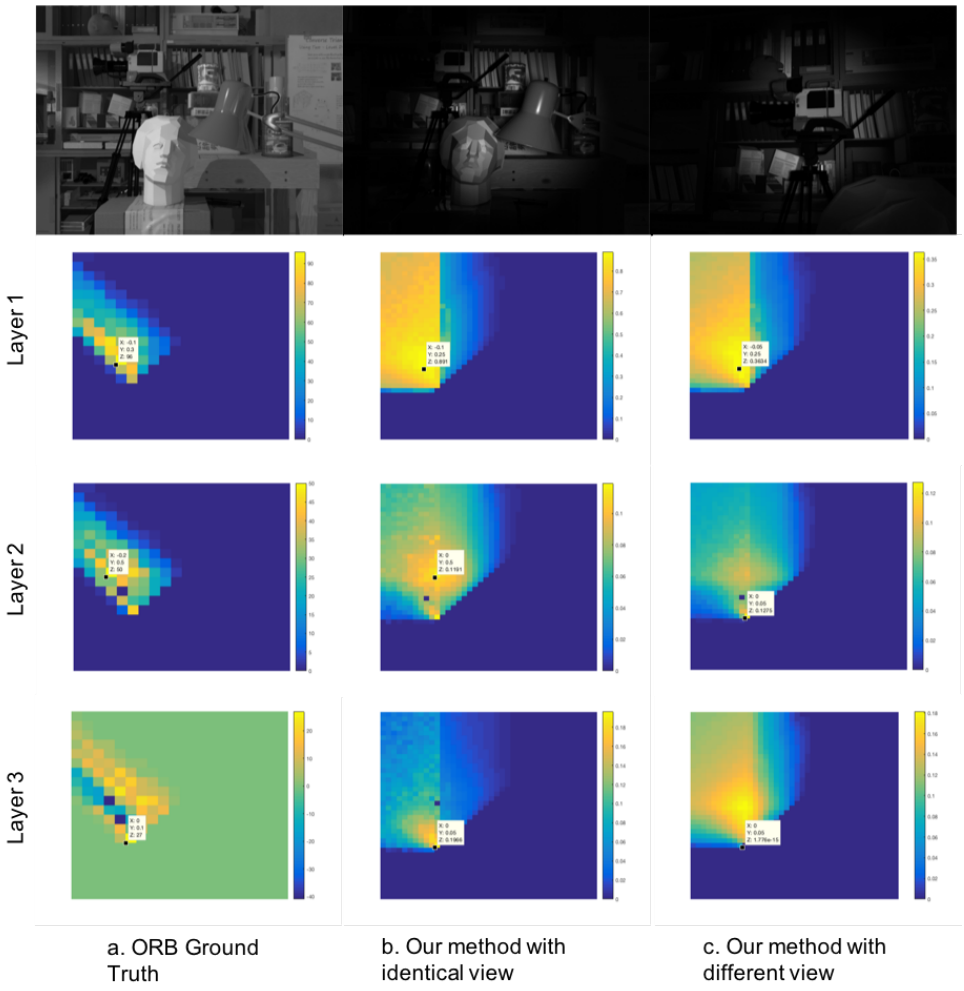


Figure 6.10 – In each layer, similar behaviors are shown with regard to optimums (see image a,b), compared with ORB detector. Ground truth for layers 2 and 3 are generated by a subtraction between the common keypoints detected from the previous optimal contrast band $\mathbf{u}^*$ and the keypoints from all others contrast bands *i.e.* the ground truth of layer 2 is computed by removing all keypoints common with keypoints detected in previous optimum $\mathbf{u}^* = (0, 0.15)^\top$ in layer 1. Empirical results also show that even with relatively different reference images (c), the landscapes are similar.

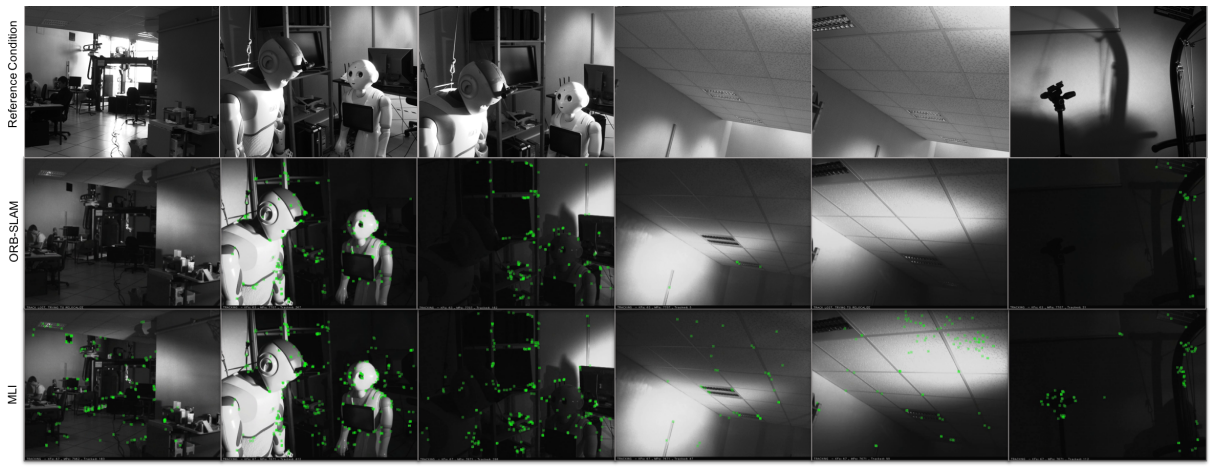


Figure 6.11 – Results of real scene against dynamic illumination variance. With a keyframe map generated under reference condition, MLI shows a better retrieve capacity especially when encountering non-uniform illumination variance. In contrast, standard ORB-SLAM only tracks the well lighted parts in the image.

virtual camera trajectories in a synthetic scene with different illumination conditions (Daylight, Fluorescent, Lamps, Flashlight).

We then designed a real scene benchmark in different static and dynamically changing lighting conditions by executing a same camera trajectory using a robotic arm (see companion videos). In both benchmarks, using a reference video in a given lighting condition, we tested the robustness of our approach compared to default ORB-SLAM in localising the camera from the second video sequence against the keyframes generated from the first video sequence, in a way similar to [99] or more recently NID-SLAM [82]. In each benchmark, we report the success rate, *i.e.* the percentage of the frames from second video successfully relocated. Our implementation is integrated in ORB-SLAM [77].

Results of NewTsukuba data set are displayed in Table. 6.3. The table reports an improved success rate against illumination changing environments in all but one condition, and provides a 96.5% success rate where both default ORB-SLAM and NID-SLAM fail (0%) in the Lamps to Flashlight condition. The reason related to the failure comparison needs to be investigated in the future work. Optimal contrast bands for the sequences are computed with a relative low sampling frequency wrt to image acquisition frequency (renew a contrast band every 5 to 10 frames).

In the case of real scenes, we placed a monocular camera on a trajectory memory 7 DoF Franka robot arm to guarantee that the camera movement in each video is identical. In comparison with the synthetic scene, a strongly dynamic lighting condition is introduced,

V_2 \ V_1	Daylight			Fluo			Lamps			Flash		
	NID	ORB	MLI	NID	ORB	MLI	NID	ORB	MLI	NID	ORB	MLI
Daylight	99.3	100	**100**	96.7	96.2	**100**	73.9	97.6	**99.7**	74.6	79.8	**90.7**
Fluo	95.0	88.1	**99.8**	99.7	100	**100**	85.3	93.9	**100**	95.8	**100**	90.5
Lamps	88.3	55.7	**99.0**	93.6	79.8	**94.1**	93.1	100	**100**	84.3	37.9	**92.4**
Flash	23.8	30.7	**92.8**	92.2	90.6	**94.6**	0.00	0.00	**96.5**	92.0	**100**	99.3

Table 6.3 – SLAM keyframe retrieval success rate among default ORB-SLAM, NID-SLAM and our MLI implementation via Mutual Information.

by having two operators randomly move spots lights in the experiment scene. Using the same *success rate* criterion, our MLI ORB implementation managed to track **100%** of the dynamically lit scene against a keyframe map generated under normal lighting condition, while normal ORB-SLAM only retrieved **52.12%** of the keyframes. Fig. 6.12 shows the inlier keypoints after graph optimization process [77], which can be regarded as trusty tracked points generated in the current frame. It demonstrates that MLI performs dramatically better than default ORB which frequently lost tracking during the video. Screenshots of the experiment video are displayed in Fig. 6.11. A better tracking quality especially around dark or over-exposed area of non-uniformly light images can be observed [1].

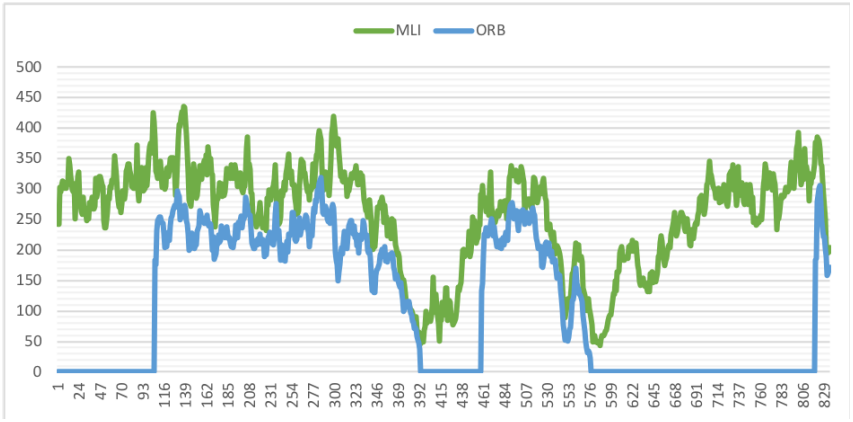


Figure 6.12 – The number of trusty inlier keypoints after graph optimization of ORB-SLAM which is a critical indicator displaying the tracking quality. MLI (green) generates significantly better results than standard ORB (blue) under the dynamic light changing environment.

1. link to video: https://youtu.be/FYuNPqFSNHw

6.9 Conclusion

Through two parts, we have introduced a novel multi-layered image representation and the computation assisted by mutual information optimization to tackle the illumination robustness problem in SLAM relocalization tasks. Each layer in MLI provides low-correlated information which helps to enhance the contrast and therefore increase the robustness during keypoints tracking process under varying illumination conditions. The optimal parameters can be searched directly in an iterative method or computed by using a multiple steps mutual information optimization framework. The proposed method shows significant improvements on both synthetic and real videos.

Chapter 7

Relative Pose Estimation and Planar Reconstruction via Superpixel-Driven Multiple Homographies

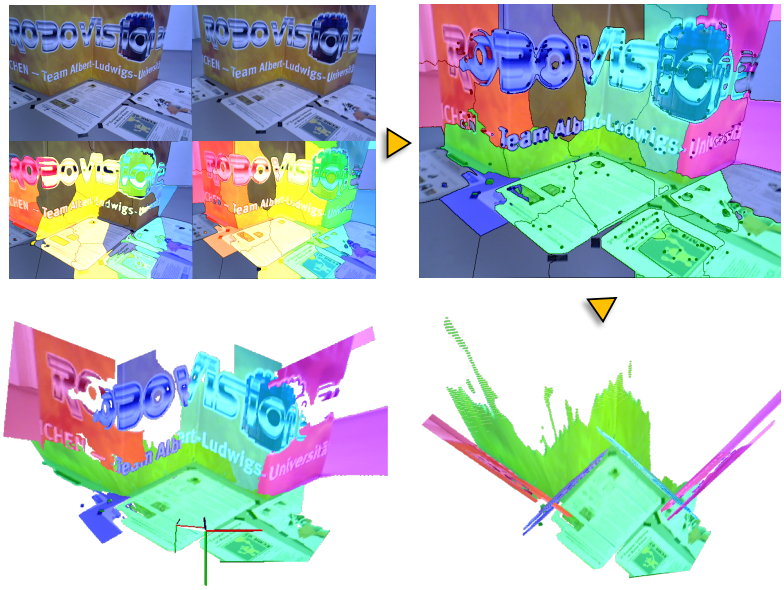


Figure 7.1 – From two RGB images of a monocular camera (left top), we propose a superpixel-driven technique to estimate simultaneously a relative camera pose and a 3D multi-planar map (bottom) without relying on a Manhattan assumption. In the right top, the different colors represent different 3D planes estimated from the images, using a novel approach we refer to as Winner-takes-all RANSAC.

7.1 Problem Description

Nowadays, many visual tracking, pose estimation and SLAM (Simultaneous Localization And Mapping) algorithms are competing to achieve better performance – precision, accuracy, computation time – in both indoor and outdoor scenarios [77, 34, 33]. Some algorithms rely on the direct alignment of the intensity between images in order to generate a dense pixel-wised mapping [34], while others exploit keypoints or similar low-level image features (*eg.* lines, patterns) to achieve more precise and robust camera poses [77]. It seems a trade-off is inevitable between the sparse methods (*eg.* keypoints-based method) and the dense methods (which compute camera poses by aligning pixel intensities): the former is more robust under variant environment and more compatible with Bundle Adjustment techniques and the latter yields a more applicable map with denser information. Though some hybrid systems are proposed to balance the advantages of both systems [63], the topic keeps attracting researchers' attention and requires further explorations.

Intermediate features extracted from images or from low-level features can also be exploited. Typically planes are ubiquitous geometric features in human-crafted environments and objects, and exhibit good characteristics for tasks such as pose estimation and visual tracking: planes are widely studied, offer a light parameterization, are robust against environmental variance w.r.t. spatially isolated keypoints, and most importantly, planes are easy to compute from image pairs via homography constraints. Many contributions also exploit planar assumptions in a variety of vision-based robotic applications [86, 29]. Homography estimation is indeed convenient and simple whilst the scene has a dominant plane such as ground or ceiling. However in the real world, the dominant plane assumption does not always hold as it can be occluded or the scene can be composed of multiple planar structures such as indoor environments or outdoor city landscapes.

In this chapter, we propose a novel multi-homography based pose estimation method via superpixel-driven RANSAC which achieves simultaneously the camera pose estimation and the dense planar mapping from a pair of color images. We also show that this method can be integrated within a vSLAM pipeline. Our contributions are: 1) a novel RANSAC technique for multiple homographies detection problem combining information from superpixels and keypoints 2) a voting-based ambiguity-free multiple homographies decomposition process for pose estimation, and 3) a non-linear optimization pose refiner for both single image and a sequence of images (vSLAM).

7.2 Specific Related Work

For the case of dominant planar scenes, [13, 109] developed visual tracking theory and applications. For example, the work of Pirchheim et al. [86] consists of a mobile AR application under the assumption of single planar homography. However, the decomposition ambiguities of the homography matrix seem difficult to resolve using merely a geometric approach [72]. Many works exploit additional information such as: a priori known geometric shapes or combining the information from IMU (Inertial Measurement Unit) not only for eliminating the ambiguities in homography but also improving the precision of pose estimation [50, 103].

Typically, the Manhattan assumption is widely exploited in planar vision tasks [103, 39, 136]. Principally the assumption is that all planes in the environment are perpendicular in 3D, such as typical buildings or standard rooms.

Many planar SLAMs and visual tracking applications exploit RGB-D cameras which are well suited for indoor-environments. By combining available depth information, Kaess [58] proposed a planar SLAM system with a quaternion formulation of 3D plane which improves convergence of optimization under RGB-D environments, and then [53] extended it to a keyframe-based dense planar SLAM with a factor graph map using incremental smoothing and mapping (iSAM). Le and Košecka [66] also combined RGB-D sensor with Manhattan Assumption.

Many contributions on plane segmentation in images are tightly associated with the *superpixel* technique. A superpixel is defined as a group of connected pixels with consistent color or intensity information. Superpixels are usually generated with segmentation methods; typical works include SLIC [1], SEEDS [122] and graph-segmentation superpixel [36].

Concha and Civera [24] are the first who proposed to exploit superpixel techniques in a SLAM system. Their approach uses a Monte Carlo ranking to achieve the correspondence and initial 3D pose of superpixels. Then an optimization is performed to refine the plane poses with an already known camera pose estimated separately from a PTAM system. In a more recent work (DPPTAM) [23] they integrate superpixel in a semi-dense tracking system. Plane estimation is achieved by RANSAC and SVD on 3D points from semi-dense tracking. A dense mapping is also designed with found superpixels information.

Inspired from [24, 23], we propose to exploit superpixels information for estimating relative camera pose and multi-planes structure simultaneously from two images (see

Fig. 8.1). Such a system requires 1) the capacity of extracting multiple planes from two images; 2) the ability of eliminating ambiguities in homography decomposition; and 3) the possibility to combine the homography representations with the optimization framework of pose estimation for better performance;

7.3 Overview

The method we propose is composed of the following modules (see pipeline in Fig. 8.2 for the overview): (a) superpixelization and tracking process: extracting and matching corresponding superpixel information from a pair of images. (b) superpixel-driven RANSAC: detecting multi-planar structures in a robust way, (c) multiple homographies decomposition: computing camera pose and eliminating ambiguities in homographies, and (d) non-linear refiner: applying a Bundle Adjustment-like optimization camera and plane refiner for both image pairs and a sequence of images. All the modules are detailed in the following section respectively.

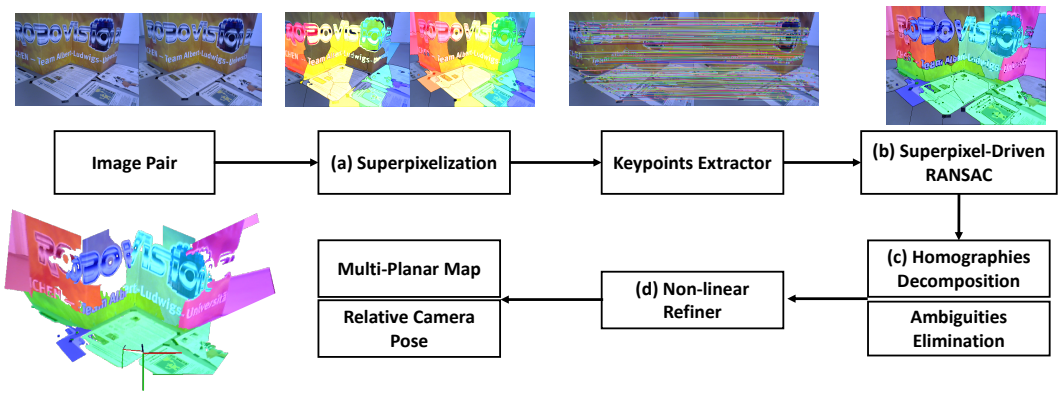


Figure 7.2 – Pipeline of our system which generates a relative camera pose and a 3D multi-planar map from a pair of color images.

7.4 Superpixel extracting and Tracking

Our work builds on the idea that superpixels are good initial guesses of planar regions in images for that they usually show strong chromatic consistency and spatial continuity at a pixel level. We exploit superpixel spatial relations (adjacency) as well as local keypoint descriptors to perform a matching of superpixels in two different frames.

More specifically, we first superpixelize two frames I_i, I_{i+1} with SLIC [1] and obtain two sets of regions respectively, denoted by $V^i = \{V_k^i\}$ with $k = 1..K$, K being the total number of superpixels extracted from ith image. We then exploit a graph structure to conserve the information of adjacency between superpixels. An unidirectional un-weighted graph is proposed: $\mathbf{G_i} = (V^i, E^i)$ where V^i the vertices are the set of superpixels in I_i and E presents their adjacency (equal to 1 when two superpixel regions are adjacent).

Once the segmentation is performed, a superpixel tracking system is required for matching superpixel regions between two frames. We undertake this step by matching keypoint descriptors (here ORB [100]) extracted from the each superpixel regions.

In contrast with common superpixel tracking tasks [125] which concentrate mostly on re-identification of moving objects from static background, SLAM and camera pose estimation works usually hold the assumption of static environment. Based on this assumption, we then propose a superpixel tracking method between two images: we search for the highest matched number of keypoints between not only two superpixel regions but also their neighbor superpixels in graph structure as in a static environment each superpixel should hold a relatively rigid local structure w.r.t others. The depth of the neighborhood d_G is represented by a on-graph distance (shortest path) used to manipulate the range of neighbor area. We denote these neighborhood regions around vertex V_k as $N^{d_G}(V_k)$, as also mentioned in Section 7.5.3:

$$N^{d_G}(V_k) = \{V_j \in \{V\} : d(V_j, V_k) \leq d_G) \tag{7.1}$$

As displayed in Fig. 7.2 and throughout the paper, matched superpixel between image pairs are highlighted with the same color.

7.5 Multi-Homography Estimation

7.5.1 Homography and RANSAC

In a planar environment, the homography matrix ${}^2\mathbf{H}_1 \in \mathbb{SL}(3)$ can be used to describe the transformation of one plane between two images I_1 and I_2. When the intrinsic calibration matrix of the camera $\mathbf{K}$ is known, all pixels extracted from I_1 and I_2 can be inversely projected as normalized three dimensional coordinates denoted as: $\mathbf{p_1}$ and $\mathbf{p_2} \in \mathbb{R}^3$. Therefore the homography matrix constrains them with the following relation:

$$\mathbf{p_2} = {}^2\mathbf{H}_1\mathbf{p_1}$$

A homography matrix is composed of a rotation matrix ${}^2\mathbf{R}_1 \in \mathbb{SO}(3)$, a translation vector ${}^2\mathbf{t}_1 \in \mathbb{R}^3$ as well as a normal vector in I_1, defined as $\mathbf{n}_1 = (a, b, c)^\top \in \mathbb{R}^3$. A plane can be therefore described as $\mathbf{p} \cdot \mathbf{n}_1 = d$, where $\mathbf{p} \in \mathbb{R}^3$ are three dimensional points on plane and d is the orthogonal distance from the plane to the origin:

$${}^2\mathbf{H}_1 = {}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1}{d}\mathbf{n}_1{}^\top \tag{7.2}$$

Multiple methods are available to compute the homography matrix ${}^2\mathbf{H}_1 \in \mathbb{SL}(3)$ from a pair of images. The *RANdom SAmple Consensus* (RANSAC) method [38] relies on two matched sets of keypoints $\{\mathbf{p_1}\}, \{\mathbf{p_2}\}$ in two frames and a Direct Linear Transform (DLT) technique [52]. Its goal is to divide the data in two sets: the set of inliers (*i.e.* Consensus-Set (CS)) and the outliers (spurious data).

We first introduce some notations used in RANSAC. We denote $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ as the set of all matched *pairs* of keypoints from I_1 and I_2: $\mathbf{x} = \{\mathbf{p_1}, \mathbf{p_2}\}$. In our case we consider the homography $\mathbf{H}$ as the model to estimate. We then define:

1. **Minimal Sample Set:** M: the minimum number of pairs of points to estimate a homography, which is 4 for one homography.
2. **Sampling Procedure**: $\mathcal{S}$: $\mathcal{D} \rightarrow \mathcal{D}^M$, it samples all subsets in $\mathcal{D}$ s.t. their cardinality equals M. The sampling is usually done by randomly selecting 4 points to compute a $\mathbf{H}$.
3. **Model Estimation Function**: $\mathcal{E}$: $\mathcal{D}^M \rightarrow \mathbf{H}$. In homography, DLT estimates $\mathbf{H}$ from 4 non-degenerated points.
4. **Inlier Threshold** ϵ: A threshold to determine inlier, here we take the distance between the point and the reprojection of it's matched pair: $(\mathbf{p_2} - {}^2\mathbf{H}_1\mathbf{p_1})^2$.

Using these definitions, one may reword the RANSAC process as an algorithm which searches for the largest Consensus-Set by randomly sampling M and evaluating their consensus via a measure function with a threshold ϵ.

7.5.2 Multi-Model RANSAC

Though RANSAC is proven to be efficient when extracting the principal plane in a scene, many applications display cases where dominant planes are occluded, and multiple

planes with similar surfaces are visible. As multiple instances of same model occur in a dataset (*eg.* multiple planes), RANSAC suffers not only from *gross outliers* (pure noise, *eg.* wrong matches of keypoints) but also from *pseudo-outliers* [112]: outliers to the structure of interest but inliers to a different structure. To solve such multi-model estimation problems (*i.e.* searching for multiple planes), many RANSAC-like algorithms have been proposed such as Sequential RANSAC [60, 123] and [143].

Sequential RANSAC consists of applying RANSAC to a multi-model dataset in an iterative fashion. For each iteration of RANSAC, the found inliers (Consensus-Set) are removed from the dataset. While the sequential nature tends to be influenced by pseudo-outliers [143], one wrong estimation of previous iteration may lead into mistakes in the following ones. To alleviate this false estimation, Kanazawa's sampling technique [60] is widely applied and proven efficient by sampling in a local proximity w.r.t the previous chosen data point (*eg.* , by Gaussian distribution) instead of randomly choosing in all dataset: $\mathbf{p} \sim \mathcal{N}(\boldsymbol{p_0}, \boldsymbol{\Sigma})$, describes the probability to choose point $\mathbf{p}$ under the condition that the previous chosen one is $\mathbf{p_0}$ and the sampling range is manipulated by Σ.

Another issue with multi-model estimation is redundancy estimation. A same model may be estimated multiple times as the inlier-removing procedure fails to totally clear out the *pseudo-outliers* of previous detected model (usually because the threshold ϵ is ill-chosen or the data experiences a heavy unbalance among different models), so the rest *pseudo-outliers* of previous model can still form a similar model which outnumbers the CS over other models. Moreover, the rest *pseudo-outliers* implicitly increases the outlier ratio along the iterations of the sequential procedure and deteriorates the estimation.

7.5.3 Superpixel-Driven Winner-Takes-All RANSAC

To address these issues, we propose a Winner-Takes-All RANSAC which is inspired by [60] but benefits from the superpixel information to address the false detection and redundancy estimation problems simultaneously. We exploit superpixels for their relative coplanarity: we assume all information inside a superpixel should be relatively coplanar, as they share local proximity and color similarity. These coplanarity regions play the role of the sampling range Σ in the Kawazana sampling. Instead of an isotropic Σ decided empirically for all datasets, we use directly the regions of superpixel as an adaptive sampling range and even avoid the computation of the conditional probability: *eg.* by only selecting points in one superpixel or its neighbor in certain on-graph distance $\mathbf{N}^{d_G}(V_k)$ (see Eq (9.1)).

We present some notations for clarity:

1. **Superpixel Cluster Map**: $\mathbf{C}$: A map returns the superpixel label from a pixel in the image. $\mathbf{C} : \Omega \subset \mathbb{N}^2 \rightarrow \mathbb{N}$
2. **Superpixel Neighbor Sampling**: $\mathcal{S}_N(\mathbf{D}, \mathbf{G}, d_G)$: A sampling method which chooses M (4 for homography) pairs of points in following way:
   (a) sample first keypoint p_1 uniformly in all dataset.
   (b) find the superpixel V_1 of p_1 via Cluster Map $\mathbf{C}$.
   (c) sample other $M-1$ points only for data in the subgraph of certain distance d_G w.r.t the V_1: $\{p_2, .., p_M\} = \mathcal{S}(\mathbf{D}(\mathbf{N}^{d_G}(V_1)))$
3. **Ratio of Inliers** ρ: two ratios are defined in this chapter, the ratio of all inliers $\bar{\rho}$ and ratio of inliers in each superpixel region ρ_k, defined as the number of inliers over the number of all the data (*eg.* extracted keypoint) and a superpixel region respectively.

The WTA-RANSAC algorithm is presented in Algo. 1. The main idea is similar to sequential RANSAC. However, after each iteration of estimation, instead of only removing CS from the dataset, we adopt a *winner-takes-all* policy: invalidate all the points in the superpixel regions where a significant higher inliers ratio shows that this superpixel is well dominated by a plane. This allows us to eliminate pseudo-outliers of the detected plane together with its Consensus-Set, as one superpixel is mainly composed by one plane, therefore improves the robustness against false and redundant estimation problem.

7.6 Homography Decomposition and Ambiguities Elimination

Once a homography matrix is found, various ways exist to decompose the ${}^2\mathbf{H}_1$ matrix to ${}^2\mathbf{R}_1$, ${}^2\mathbf{t}_1/d$, and $\mathbf{n}_1$ (in monocular case, the translation vector is up to a scale factor). Analytically, linear decomposition methods performs well yet generate some ambiguities. Two ambiguities exist even after applying the condition which all points are visible to the camera. Ambiguity can be solved if at least one element among $\mathbf{R}, \mathbf{t}, \mathbf{n}$ is known a priori, *eg.* , the normal direction of the floor is known as perpendicular to the up direction, or an IMU is able to indicate the direction of the motion or other measure methods to filter the ambiguity resu1ts.

Algorithm 1: Algorithm Winner-Takes-All RANSAC

Data: $\mathbf{D}$, ϵ, M, $\mathbf{G}$, $\mathbf{C}$, q, d_G
// q a parameter controls the level of WTA
Result: S_H
$S_H = \{\}$ // the set of multiple H ;
$S_{ov} = \{\}$ // indicate the occupation of each vertex **while** *!StopCondition* **do**
  // single iteration of RANSAC ;
  for *iterations* **do**
    $M = \{\mathcal{S}_{\mathcal{N}}(\mathbf{D}, \mathbf{G}, d_G) : C(p) \notin S_{ov}\}$;
    $H = DLT(M)$ // estimate H ;
    $CS = \{p \in D : E(H, p) < \epsilon, C(p) \notin S_{ov}\}$;
    if $(|CS| > MaxCS)$ **then**
      $BestH, MaxCS = H, |CS|$;
    end
  end
  // Winner-takes-all ;
  for $V_j \in V(G)$ **do**
    if $(\rho_j > q \times \bar{\rho})$ **then**
      $S_{ov} = S_{ov} \cup j$;
    end
  end
  $S_H = S_H \cup BestH$;
end

The main reason of the impossibility in differentiating two ambiguities is that geometrically both of them hold the homography constraint. In the work of [72], the relation of the translation vector between these two ambiguities $\{\mathbf{R}_a, \mathbf{t}_a, \mathbf{n}_a\}$ and $\{\mathbf{R}_b, \mathbf{t}_b, \mathbf{n}_b\}$ is displayed as follows: (for simplicity and under the circumstance of no confusion, we abuse the notation of $\mathbf{R}_a$ to describe ambiguities ${}^2\mathbf{R}_{1a}$ in this section; this is similar for all other notations)

$$\mathbf{t}_b = \frac{\|\mathbf{t}_a\|}{\rho}\mathbf{R}_a\left(2\mathbf{n}_a + \mathbf{R}_a^\top \mathbf{t}_a\right) \tag{7.3}$$

$$\rho = \left\|2\mathbf{n}_e + \mathbf{R}_e^\top \mathbf{t}_e\right\| > 1; \quad e = \{a, b\} \tag{7.4}$$

Eq. (7.3) and (7.4) show that the difference between $\mathbf{t}_a$ and $\mathbf{t}_b$ is actually influenced by $\mathbf{R}_a$ and $\mathbf{n}_a$. For a case with a single homography, one cannot exploit this relation for selecting a *true* transformation between two images. However, under the condition of the multiple homographies, the Eq. (7.3) is applied with an extra constraint. All the homographies actually share a common translation and rotation across different planes, as the scene is static while the camera is moving. Our intuition is then to rely on this shared information to eliminate the decomposition ambiguities.

For each $\mathbf{H}^i$ in the multiple homography scene $\{\mathbf{H}^i\}$, two possible ambiguities can be expressed as the ground truth set $\{\mathbf{R}_t{}^i, \mathbf{t}_t{}^i, \mathbf{n}_t{}^i\}$ and its ambiguity set $\{\mathbf{R}_f{}^i, \mathbf{t}_f{}^i, \mathbf{n}_f{}^i\}$. As all homographies share a unique $\mathbf{t}_t$ and $\mathbf{R}_t$:

$$\mathbf{t}_f^i = \frac{\|\mathbf{t}_t\|}{\rho}\mathbf{R}_t\left(2\mathbf{n}_t^i + \mathbf{R}_t^\top \mathbf{t}_t\right) \tag{7.5}$$

This means the relation between the real translation $\mathbf{t}_t$ and the ambiguous one $\mathbf{t}_f{}^i$ is only influenced by the normal vector of the plane $\mathbf{n}_t{}^i$. Under the assumption that at least two planes have different normal vectors (which is very common is the multiple planar scene), one could find the real transformation $\{\mathbf{R}_t, \mathbf{t}_t\}$ by simply choosing the common translation vector, and therefore eliminate the ambiguity solutions to the unique one. This procedure is performed by implementing a fairly straightforward voting system on the direction of all translation vectors. By accounting for an angle threshold δ ($15°$ in our implementation) to gather vectors, we select the most voted translation vector and therefore eliminate the ambiguities of each plane.

7.7 Non-linear Multi-Plane Refiner

7.7.1 Non-linear Refiner of Image Pair

In traditional SLAM systems, Bundle Adjustment techniques are introduced to refine camera poses and landmarks by minimizing the re-projection error on image space of landmarks such as keypoints, lines or other features. Likewise, for the case of homography transformation between two images, previous work (*eg.* image-based visual servoing system [73]) have already shown that with a prior known plane, the estimation of the camera pose $\mathbf{q} \in \mathfrak{se}(3) \in \mathbb{R}^6$ (the minimal representation of transformation $\{\mathbf{R}, \mathbf{t}\}$) can be realized via a least square Gauss-Newton optimization process by similarly minimizing the re-projection error E between extracted $(\mathbf{p}_2^n - {}^2\mathbf{H}_1\mathbf{p}_1^n)^2$ being $n = 1..N_p$ as number of keypoints. By now adding the plane parameter $\mathbf{\Pi}_1 = \{\mathbf{n}_1, d\}$ into the system, for a single homography, the optimization framework has the following form:

$$\{\widehat{\mathbf{q}}, \widehat{\mathbf{\Pi}_1}\} = \underset{\mathbf{q},\mathbf{\Pi}_1}{\operatorname{argmin}} \; E(\mathbf{q}) = \underset{\mathbf{q},\mathbf{\Pi}_1}{\operatorname{argmin}} \sum_n^{N_p} \left(\mathbf{p}_2^n - {}^2\mathbf{H}_1\mathbf{p}_1^n\right)^2 \tag{7.6}$$

In a dense form the Jacobian of Eq. (8.4) can then be reformulated as:

$$J(\mathbf{q}, \mathbf{\Pi}) = \begin{bmatrix} \frac{\partial E}{\partial \mathbf{q}} & \frac{\partial E}{\partial \mathbf{\Pi}} \end{bmatrix} \in \mathbb{R}^{2\times 10} \tag{7.7}$$

With the Jacobian of camera pose $J(\mathbf{q})$ defined as the Jacobian of $E(\mathbf{q})$ in $\mathbf{q}$:

$$J(\mathbf{q}) = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{7.8}$$

where (x, y) are 2D points coordinates corresponded to $\mathbf{p}$, $1/Z$ is the inverse depth and computed as follows with $\mathbf{p_2}$ keypoint in frame 2 (see [73]):

$$1/Z = \frac{d - {}^2\mathbf{t}_1\mathbf{n}_1}{{}^2\mathbf{R}_1\mathbf{n}_1\mathbf{p}_2} \tag{7.9}$$

Similarly for Jacobian of plane $\frac{\partial E}{\partial \mathbf{\Pi}}$, four columns representing $\frac{\partial E}{\partial n_x}$, $\frac{\partial E}{\partial n_y}$, $\frac{\partial E}{\partial n_z}$, $\frac{\partial E}{\partial d}$.

$$J(\mathbf{\Pi}) = \begin{bmatrix} \frac{x(t_z x - t_x)}{d} & \frac{y(t_z x - t_x)}{d} & \frac{(t_z x - t_x)}{d} & \frac{1/Z(t_z x - t_x)}{d} \\ \frac{x(t_z y - t_y)}{d} & \frac{y(t_z y - t_y)}{d} & \frac{(t_z y - t_y)}{d} & \frac{1/Z(t_z y - t_y)}{d} \end{bmatrix} \tag{7.10}$$

t_x is the x axis value in $\mathbf{t} = (t_x, t_y, t_z)^\top$.

However, for the case of multiple homographies in a static environment between two

images, the relation of a set of homographies detected in the image $\{^2\mathbf{H}_1^i\}$ consists of a shared transformation $^2\mathbf{R}_1, ^2\mathbf{t}_1$, where $i = 1..N_{\mathbf{\Pi}}$ as the number of plane:

$$^2\mathbf{H}_1^i = {}^2\mathbf{R}_1 + \frac{^2\mathbf{t}_1}{d_i} {\mathbf{n}_1}^{i\top} \tag{7.11}$$

By exploiting this characteristic, we propose a camera pose and plane refiner for multiple homographies:

$$\{\hat{\mathbf{q}}, \{\widehat{\mathbf{\Pi}_1^i}\}\} = \underset{\mathbf{q},\mathbf{\Pi}_1^i}{\operatorname{argmin}} \sum_i^{N_{\mathbf{\Pi}}} \sum_n^{N_p} \left(\mathbf{p}_2^n - {}^2\mathbf{H}_1^i \mathbf{p}_1^n\right)^2 \tag{7.12}$$

The Jacobian actually holds a sparse form, for example the block of Jacobian for computing all keypoints in plane $i \in 1 \dots N_{\mathbf{\Pi}}$ can be then defined as:

$$J(\mathbf{q}, \mathbf{\Pi}^i) = \left[\frac{\partial E}{\partial \mathbf{q}} \underbrace{0 \cdots 0}_{4(i-1)} \frac{\partial e}{\partial \mathbf{\Pi}^i} \underbrace{0 \cdots 0}_{4(N_{\mathbf{\Pi}}-i)} \right] \in \mathbb{R}^{2\times(6+4N_{\mathbf{\Pi}})} \tag{7.13}$$

Therefore the Jacobian of single image refiner for all planes is:

$$J(\mathbf{q}, \mathbf{\Pi}) = \left[J(\mathbf{q},\mathbf{\Pi}^0)^\top \; J(\mathbf{q},\mathbf{\Pi}^1)^\top \; \dots \; J(\mathbf{q},\mathbf{\Pi}^{N_{\mathbf{\Pi}}})^\top \right]^\top \tag{7.14}$$

Refer to Section 8.7 for the visualization of estimation between image pairs.

7.7.2 Bundle Adjustment-like Refiner

Plane Association

Unlike keypoint-based Bundle Adjustment (BA) techniques widely used in [77][62], our 3D planar map is designed as a two-level structure: extracted keypoints belong to different planes respectively. Therefore a plane association process is mandatory for the following BA section. The problematic can be reformulated as follows: we search for a way of matching two sets of planes from two frames respectively $\{\mathbf{\Pi}_c\}$ and $\{\mathbf{\Pi}_{c+1}\}$.

In contrast with related work which directly compare these plane parameters $\{\mathbf{n}, d\}$ without considering image information [58], or others which only consider image overlapping information but do not account for geometric constraints, we propose a hybrid plane association policy considering both geometric and on-image information:

i) As the distance d is heavily influenced by scale ambiguity we first compare the angle between two normal vectors $d(\mathbf{n}_c, \mathbf{n}_{c+1})$. However this method does not differentiate two

parallel planes in the environment.

ii) Superpixel tracking results are also taken into consideration. It not only helps avoid the parallel plane from mismatching but can also reject the camera pose when the translation is too small between images and all planes become one homography.

iii) We finally check the number of matched descriptors among planes. A window search after re-projecting by homography can also be applied for a more robust matching result: *eg.* for comparing the keypoints between frame p_{c+n} and frame p_c, as no direct ${}^{c+n}\mathbf{H}_c$ computed from image, we can simply propagate the keypoints in frame i by multiplying the homography matrices: ${}^{c+n}\mathbf{H}_{c+n-1} \dots {}^{c+1}\mathbf{H}_c p_c$ and compare them with p_{c+n} in a window searching method.

Plane Map Refiner

The Plane Map refiner consists in an optimization framework which refines all keyframes' poses and their common planes thanks to our plane matching process. Each keyframe contains multiple planes and keypoints in each plane. Once the *joint plane* information is gained over different keyframes, like global BA for point-based SLAMs, this procedure eliminates the drifting problem, solves scale ambiguity and refines camera trajectory w.r.t whole sequence. The BA-like optimization approach we propose accounts for all homographies from all different keyframes:

$$\arg\min_{\mathbf{q}_c, \mathbf{\Pi}_c^i} \sum_{c}^{N_c} \sum_{i}^{N_\Pi} \sum_{n}^{N_p} \left(\mathbf{p}_{c+1}^n - {}^{c+1}\mathbf{H}_c^i \mathbf{p}_c^n \right)^2 \tag{7.15}$$

where c is the index of the frame number and i is the index of plane number, N_c and N_Π represent the total frame number and plane number respectively.

Keyframe Selection

Our proposed keyframe selection is a straightforward heuristic comparable to systems like [77, 34]. We rely on the parallax metric (defined as an average translation of all matched keypoints between images) and matching quality for choosing keyframes. Two conditions are checked i) to have a parallax on at least a given number of pixels; this is a hyper-parameter from one dataset to another, empirically found between 20 to 40 pixels, and ii) at least a certain number of planes is well matched. This parameter is also adjustable as some environments include many small planes and some comprise less.

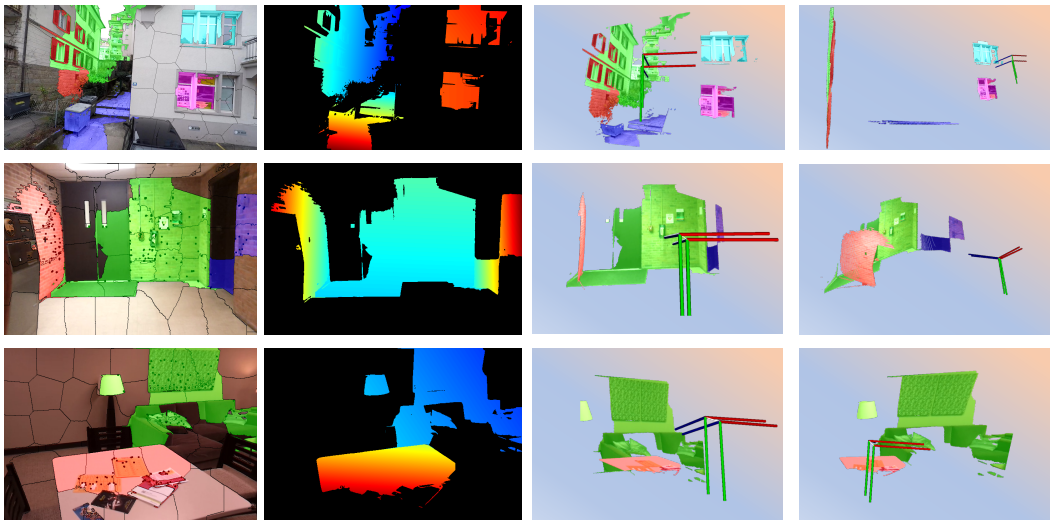


Figure 7.3 – Demonstration of results estimated from image pairs. Depth image and 3D planar maps are also illustrated showing that our method estimates well under the multi-planar environment. Result (c) shows that our method conserves well the orthogonality among planes without relying on the Manhattan assumption.

7.8 Experiments

Our experiments include three parts: image pairs, indoor experiment and outdoor experiment.

We test various image pairs under different environment and camera types across a wide range of datasets including RGB image of Kinect camera [113], hand-held mobile phone [43] and Micro Air Vehicle images [71]. Results are presented in Fig 7.3 with the plane estimation, correspondent depth image as well as a 3D planar map with camera pose. Another example of comparison is given in Fig 7.4, the estimated depth image corresponds well to the ground truth estimated by Kinect camera and is able to keep a very dense form which seems difficult for sparse and even semi-dense RGB monocular mapping systems.

To test indoor environment on whole image sequences, we relied on the TUM RGB-D dataset[113] also used in [23, 48]. The scene is constructed as a pure planar environment, however the homogeneous color distribution on the pop-up shape wall is relatively challenging for superpixel extraction: many superpixels are spawned at the frontier of two planes as their color seems very similar. See Table I for the generated results by comparing with ORB-SLAM [77], LSD-SLAM [34], Multi-Level Mapping [48] and DPPTAM [23].

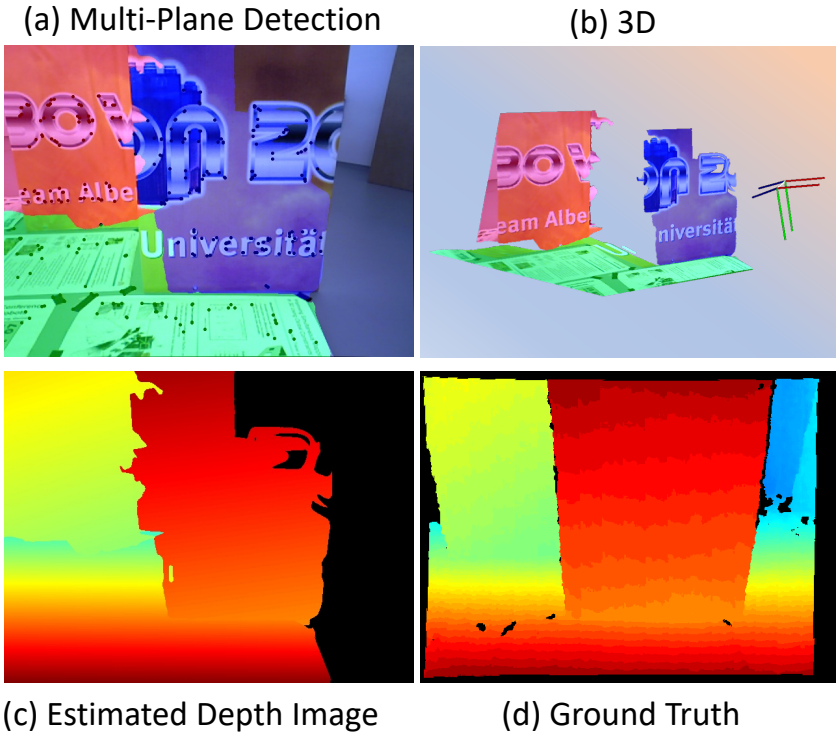


Figure 7.4 – Comparison of estimated results from *image pair* against the depth map from ground truth on the dataset TUM [113]. With a small number of parameters (3 planes), our proposed method is able to generate a very dense map.

	ATE (m)		
Methods	***Mean***	***Median***	***RMSE***
ORB-SLAM	**0.010**	**0.009**	**0.012**
LSD-SLAM	0.157	0.124	0.170
Multi-Level Mapping	-	-	0.17
DPPTAM	0.063	0.063	0.065
Well Selected KF (ours)	**0.023**	**0.017**	**0.027**
Mean (ours)	0.037	0.031	0.045
Median (ours)	0.040	0.029	0.047

Table 7.1 – Evaluation of absolute trajectory error (ATE) against different methods. The proposed method outperforms DPPTAM, LSD-SLAM and Multi-Level Mapping. Despite behind ORB-SLAM performance (a keypoint-based SLAM technique that generates a sparse point cloud map), our approach provides a dense map representation (mean and median results are computed on five consecutive runs).

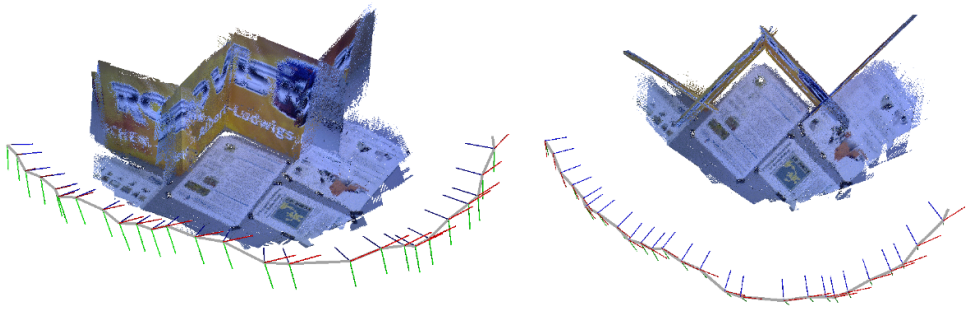


Figure 7.5 – 3D multiple plane map and camera trajectory of the dataset TUM [113] generated by our method.

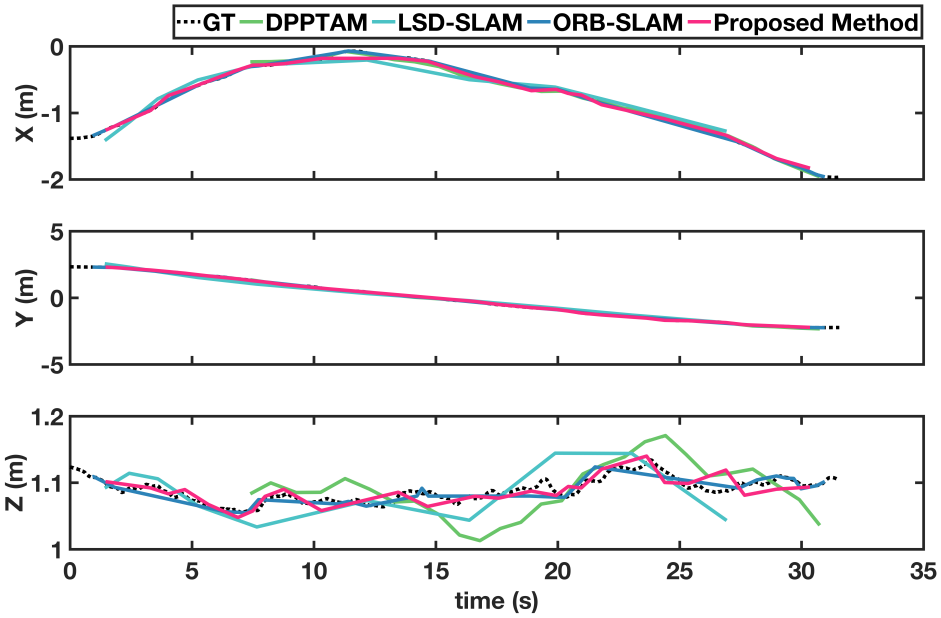


Figure 7.6 – Comparison of trajectories generated from different methods: Our proposed method shows a more stable and similar trajectory results w.r.t LSD-SLAM and DPPTAM, reaches the save level of state-of-the-art sparse SLAM method ORB-SLAM, thanks to the global planar representation and non-linear BA.

Our method outperforms all dense and semi-dense methods in terms of absolute pose error (ATE) and reaches a good level of precision against a state-of-the-art monocular sparse keypoint-based SLAM [77] which only provides sparse point cloud mapping.

Finally we test our system on image sequence from hand-held monocular gray-level camera dataset [35], under an outdoor and corridor-like environment. Fig 7.7 displays that our system successfully recovers the multiple planes structure as well as a camera

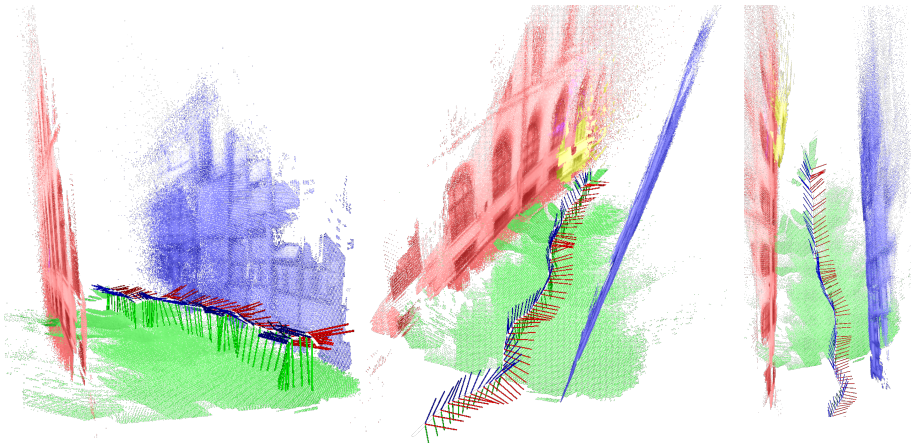


Figure 7.7 – Experiment on an outdoor dataset [35], coordinates represent the camera pose of keyframes. The multi-planar structure is well conserved without applying any assumptions under a corridor-like environment.

trajectory from the sequence. [1]

7.9 Conclusion

We proposed a novel method to estimate a camera pose from sparse keypoints and simultaneously reconstruct a dense planar map representation via multiple homographies. A superpixel-driven RANSAC method was introduced to perform multiple homography extractions from planes, and homography ambiguities were resolved using a voting system. We also introduced an optimization camera and plane map refiner to perform more precise mapping and tracking results. Results demonstrate the benefits of the approach in comparison with existing contributions.

Future work will focus on improved plane matching techniques and life-long performance, to match precision of sparse SLAM techniques, and yield more lightweight map representations than dense SLAM techniques.

1. link to video: https://youtu.be/Q9L4O7hK3ME

Chapter 8

TT-SLAM: Dense Monocular SLAM for Planar Environments

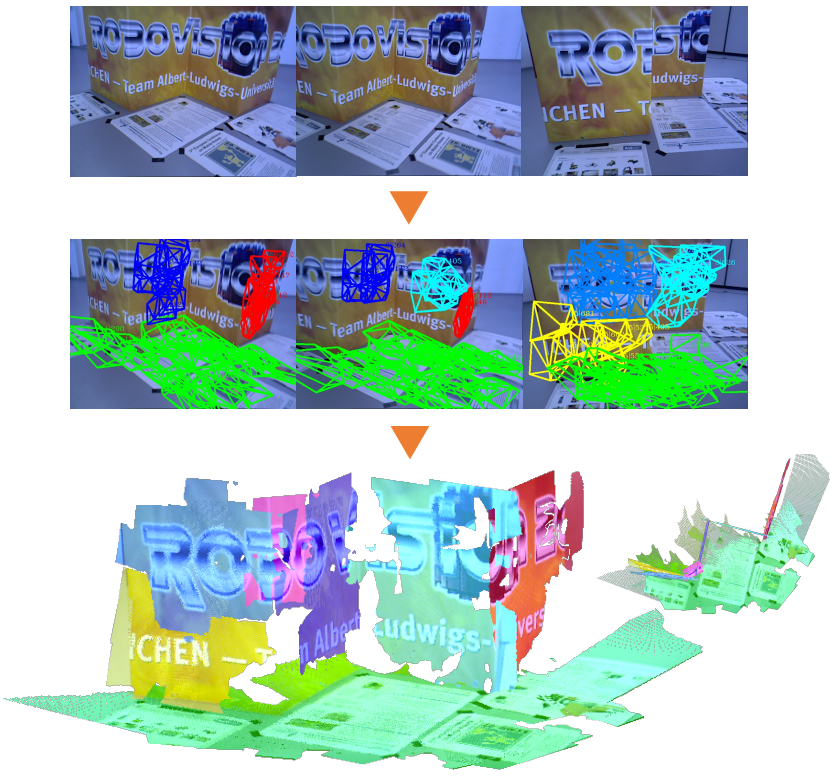


Figure 8.1 – We propose a method which tracks and clusters template-based trackers, estimates camera pose and maps three dimensional multi-planar environment on RGB image sequence acquired by monocular camera. Different colors represent different found three dimensional planes.

8.1 Problem Description

In previous chapter, we presented a system which can exploit mulitple homography structure to estimate relative camera pose and map multiple planar environment simultaneously, based on RANSAC technique with the help superpxiel. The advantages consists of using less parameterization for undertaking the reconstruction in dense fashion.

In this chapter, we follow the same problem definition but relying on a different computer vision concept: template trackers, we propose a multiple planar SLAM framework using template-based trackers and superpixels to estimate camera trajectories and reconstruct dense mapping from monocular image sequences (Fig.8.1).

Our contributions are: 1) A novel method of initializing template trackers with the help superpixels. 2) A system based on meanshfit clustering to handle the planar segmentation and pose estimation. 3) A framework of merging template tracker estimation into a nonlinear optimization refiner for improving the precison and robustness.

8.2 Specific Related Work

Template-based trackers are created to track and estimate planar image patches by registering different primitive geometric models w.r.t various metrics: *eg.* sum of square difference (SSD), zero-mean normalized cross-correlation (ZNCC), and even mutual information (MI). Plane Trackers usually estimate a homography transform between template patch and query image via optimization method. Many applications are derived from template-based trackers including: augmented reality [73], robot control [111], etc. Comparing with RANSAC methods (*eg.* [130]), using template tracker to extract homographies continuously has following advantages: 1) It solves well the data association problem when multiple planes present in the scene; 2) It provides continuous observation of the tracking results, therefore the system has more flexibility to deal with the keyframe selection problem; 3) RANSAC method tends to require higher computational cost, as the template trackers are much lighter and determinist in terms of yielded results.

Combining the advantage of template tracker and the work of multiple homographies pose estimation in previous chapter [130], we present a novel method of multiple planar vSLAM with help of template trackers. It supports: 1) a novel method of tracking camera pose and mapping multiple planar environment simultaneously in dense fashion; 2) a framework of generating, clustering and utilizing template trackers with support

of superpixel images for vSLAM applications; 3) a mean of applying homography-based non-linear optimization on template trackers for achieving better pose estimation and mapping quality

8.3 Overview

We propose a method for doing vSLAM with help of template trackers. It comprehends following modules (see overview pipeline in Fig. 8.2): (a) generation and tracking of template trackers: we add template trackers on region of superpixelizd images and track them in image sequence. (b) Clustering of decomposed planes: we rely on Meanshift clustering algorithm for grouping similar planes decomposed from found homographies to extract multiple planar structure (c) non-linear refiner: applying a non-linear optimization framework on template trackers for refining camera pose and multiple planes simultaneously on both single incoming image and whole image sequence (Bundle Adjustment-like). All the modules are detailly discussed in the following sections:

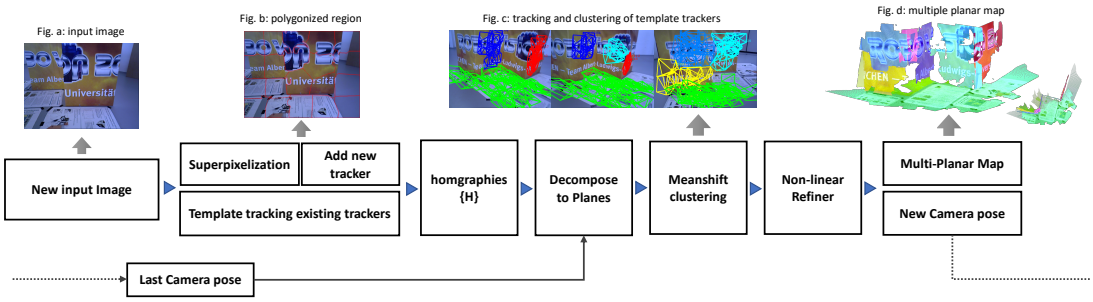


Figure 8.2 – Pipeline of our system which process input image sequence (subfig.a) to superpixelization (subfig.b). In subfig.c, tracking and clustering template trackers are undertaken (different color represent different found planes in 3D, see subfig.d). Pass through the module of refiner, our method is able to recover camera trajectories and a dense planar environment which conserves well perpendicularity without applying Manhattan assumptions.

8.4 Multiple Template Trackers

The intuition behind our work is to exploit multiple template-based trackers to estimate camera pose from multiple planar scene regions.

Planar template tracker is a technique which tracks planar image regions for a sequence of frames. It outputs homography transform $\mathbf{H}$ from region of the initialized image to the current image frame. In planar scenes, the homography transform ${}^2\mathbf{H}_1 \in \mathbb{SL}(3)$ is used to describe the transformation of one three-dimensional plane from one image frame I_1 to another I_2. When the camera is intrinsically calibrated, *i.e.* the intrinsic matrix $\mathbf{K}$ is known, all pixels from I_1 and I_2 can be presented as normalized three dimensional coordinates denoted as: $\mathbf{p_1}$ and $\mathbf{p_2} \in \mathbb{R}^3$. The homography matrix is therefore a constraint between those points within the planar region:

$$\mathbf{p_2} = {}^2\mathbf{H}_1\mathbf{p_1}$$

This matrix is actually composed of a rotation matrix ${}^2\mathbf{R}_1 \in \mathbb{SO}(3)$, a translation vector ${}^2\mathbf{t}_1 \in \mathbb{R}^3$ and a normal vector in first frame I_1: $\mathbf{n}_1 = (a, b, c)^\top \in \mathbb{R}^3$ (Eq. 8.1). The three dimensional plane associated is then formulated as $\mathbf{p}^\top\mathbf{n}_1 = d$, where $\mathbf{p} \in \mathbb{R}^3$ are three dimensional points on the plane and d is the perpendicular distance from the plane to the origin:

$${}^2\mathbf{H}_1 = {}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1}{d}{\mathbf{n}_1}^\top \tag{8.1}$$

Various methods are invented to compute a homography matrix between images, some rely on the keypoints [60] and others exploits pixel level information [2]. For most of template tracking problem, it is regarded as a differential image alignment problem on pixel-level.

The objective of differential image alignment is to estimate a displacement ρ of an image template I^* in multiple frames. It can be treated as a frame-to-frame tracking process, where the I^* is usually a Region-of-Interest (RoI) extracted from the initialized frame, in our case the region of superpixel generated from initialization frame. Besides, one requires a similarity measure f to represent the distance between reference image and wrapped image. With above definitions we may describe the differential image alignment problem under an optimization framework:

$$\widehat{\rho}_t = \arg\max_{\rho} f\left(I^*, w\left(I, \rho\right)\right) \tag{8.2}$$

where we aim to find the displacement $\widehat{\rho}_t$ that maximizes the similarity under a given metric f. For the purpose of clarity, the warping function w is an abused notation to define a general transformation of the image I paramterized by ρ. In the circumstances of

planar homography estimation, we search on $\rho \in \mathfrak{sl}(3)$ which has 8 parameters. In order to accelerate the searching process, inverse compositional formulation technique is proposed by precomputing derivatives of the reference image (see more details in [10, 29]). In this chapter, the implementation is achieved by using the inverse compositional template-based visual trackers from ViSP library [111] equipped with different similarity measures: sum of square difference (SSD), zero-mean normalized cross-correlation (ZNCC), and even mutual information (MI) [29].

Unlike the common applications of template-based trackers, where the regions-of-interest are usually known in a priori, or even selected by user interaction. The first point our system needs to address is to generate adequate regions in terms of their area and location consistent with rough planar assumption. To solve this problem, we apply the superpixel technique for generating these regions. Superpixel is defined as a group of connected pixels sharing strong chromatic consistency (*eg.* SLIC [1]). One assumption is made here is that each superpixel can be regarded as a potential planar region tractable by template-based trackers.

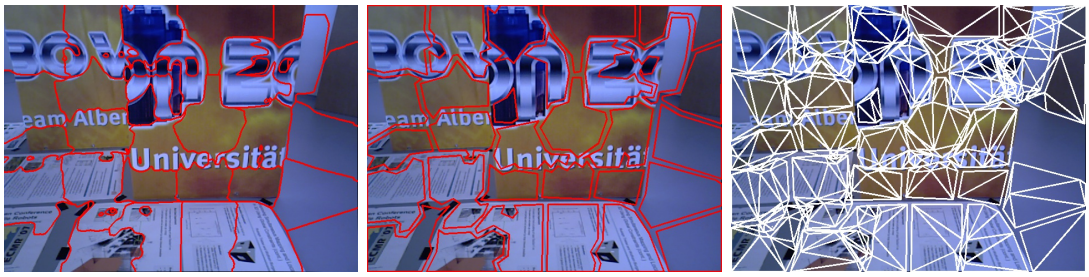


Figure 8.3 – An example of template tracker generation process. In the left subfig shows the cluster contour of superpixelized image. Polygonized region and spawned template trackers with triangulized RoIs are demonstrated in the middle and right subfigs respectively.

Therefore during the initialization procedure, each superpixel is assigned to a template-based tracker as RoI for tracking them in the following frames. Technically, we extract simplified the contour of superpixels by applying Teh-Chin chain approximation [116] and Ramer–Douglas–Peucker algorithm [91] on eroded superpixel contour as superpixel borders are often non-planar and pertubate tracking quality. Finally the regions are represented as Delaunay triangulation as valid tracking RoIs (See Fig.8.3). Though the superpixels only assure a rough a priori of planar region, theoretically, trackers which are assigned with non-planar region would soon lead to the unconvergence during the tracking

optimization process and then can be removed from the system.

Different from our previous work [130], where all homographies are estimated from one certain keyframe (*i.e.* the same reference image), the template trackers can be added from any frame for achieving a less keyframe-dependent performace. We add template trackers on the newly generated superpixels of incoming image which fail to superpose by an already exist template tracker via measuring their ratio of coverage on image surface. In another word. For every new incoming frame, we compare the yielded superpixels with current valid trackers and add new trackers on which are not covered. The ratio is defined as follows for each superpixel:

$$r = \frac{S_{tt} \cap S_{sp}}{S_{sp}} \tag{8.3}$$

S_{tt} and S_{sp} are the regions of template tracker and superpixels respectively.

8.5 Clustering and Decomposition

Once we achieve a set of homography from template trackers $\{\mathbf{H}\}$ in a multiple planar scene, next step is to extract multiple planar structure from each tracker's output (*i.e.* homography), in other word, we would like to know if multiple trackers belong to the same plane structure so that we may exploit this common information for better estimation and simplified representation. In the previous work [130], this process is done by a proposed Winner-Takes-All RANSAC on detected keypoints. As we do not utilize keypoints but template trackers in this chapter, we propose to utilize clustering techniques for handling this classification problem to decide which trackers belong to a same plane.

Clustering is a task of grouping similar data together and doing the classification according to specific metrics: classic works including K-means [69], meanshfit [42]. It's very popular in computer vision and visionary robotics application as it's able to reveal patterns of system from data aspect: *eg.* [61] use Meanshift technique for estimating undrifted rotation from vanishing points in indoor scenarios to decouple the rotation and translation in SLAM.

In this chapter, we expect a clustering system to separate different trackers and group the similar ones as they are tracking the same three dimensional plane. As we do not know in advance the quantity of planes in the scene, it makes Meanshift an appropriate method to deal with the case as it doesn't require intial seed number unlike many clustering

methods. Ideally, if all the trackers are initialized at the same reference frame, we may directly apply the Meanshift on the space of homography $\mathbf{H} \in \mathbb{SL}(3)$. However, as aforementioned adding tracker policy, it seems infeasible to undertake the classification directly on the homography space for that we are dealing with trackers initialized from different reference frames. Instead, given the nature of pose estimation is a sequence tracking problem, another solution is classifying on the decomposed planes represented in world coordinate (see Eq. 8.1), and clustering them in the space of plane parameters $\mathbf{\Pi} = \{\mathbf{n}, d\}$ where $\mathbf{n}$ is the normal vector of the plane and d is the perpendicular distance to the origin.

Another problem during the decomposition is the ambiguity problem: Inevitably, decomposing single homography yields two sets of result of $\mathbf{R}, \mathbf{t}, \mathbf{n}$ which both of them are geometrically valid. Without extra information, at least two ambiguities exist even after applying positive depth condition, unless one element among $\mathbf{R}, \mathbf{t}, \mathbf{n}$ is known in a priori: *eg.* by IMU information or known surface normal like ground or wall. For multiple planar homographies, it addressed in a previous chapter [130], by voting on the common direction of translational vector. We adopt the same method in this chapter for not only eliminating ambiguities but also filtering low quality template trackers by measuring their translational vector to the voted common direction: if none of translational vectors is close enough to the common direction among ambiguity sets, we can say that the template tracker itself may be wrongly initialized or assigned with non-planar regions.

After decomposition, a set of planes represented in world coordinate can be denoted as $\{\mathbf{\Pi}\}$. In order to achieve the classification, instead of clustering naively on the space of plane $\mathbf{\Pi} = \{\mathbf{n}, d\}$ where the euclidean distance not defined properly, a hierarchical Meanshift scheme is applied on first layer the normal vectors $\{\mathbf{n}\}$ and then second layer the d parameter and the on image barycenter position of region $\{d, p_c\}$ of each template trackers for grouping planes locally consistent. We utilize the euclidean metric on both hierarchies of clustering and find the results are good enough though the normal space has its own geodesic metric on Sphere group.

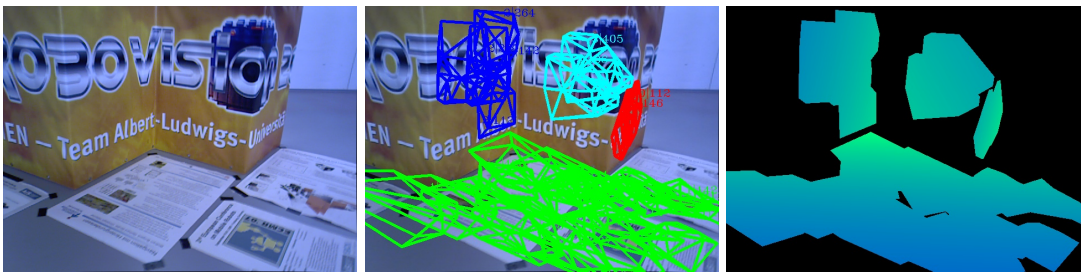


Figure 8.4 – Clustered and matched template trackers (middle) and correspondent generated depth at trackers region (right).

8.6 Non-linear Multi-Plane Refiner and BA

8.6.1 Non-linear Refiner of Current Image

As long as we know the classification results of template trackers, it requires a refining process for exploiting multiple trackers information and estimating camera pose $\mathbf{q} \in \mathfrak{se}(3) \in \mathbb{R}^6$ (the minimal representation of transformation $\{\mathbf{R}, \mathbf{t}\}$) and planar equation $\mathbf{\Pi}$ simultaneously. In traditional SLAM systems, this process is usually handled by non-linear optimization frameworks, which minimize the re-projection error on image space of extracted landmarks such as keypoints (Bundle Adjustment).

For homography transformation, similar framework can be achieved via a non-linear least square Gauss-Newton optimization process by also minimizing the re-projection error E between pixels $(\mathbf{p}_2^n - {}^2\mathbf{H}_1\mathbf{p}_1^n)^2$, $n = 1..N_p$ as number of pixels, w.r.t. camera pose $\mathbf{q}$ and the plane parameter $\mathbf{\Pi}_1 = \{\mathbf{n}_1, d\}$, it can be rewritten as the following form:

$$\{\widehat{\mathbf{q}}, \widehat{\mathbf{\Pi}_1}\} = \underset{\mathbf{q},\mathbf{\Pi}_1}{\operatorname{argmin}} \; E(\mathbf{q}) = \underset{\mathbf{q},\mathbf{\Pi}_1}{\operatorname{argmin}} \sum_{n}^{N_p} \left(\mathbf{p}_2^n - {}^2\mathbf{H}_1\mathbf{p}_1^n\right)^2 \tag{8.4}$$

The analytic Jacobian of this optimization can be found in [130]. However, different from [130], no presence of keypoints are available in template trackers. Therefore, we utilize the corners of triangulation process as keypoints for representing the found homography by template trackers.

Similarly to [130], the case of multiple homographies in a static environment can be reinterpreted as the relation of a set of homographies estimated by trackers $\{\mathbf{H}^i\}$ and a shared transformation in world coordinate frame ${}^w\mathbf{T}_o$ represented by local transforms ${}^w\mathbf{T}_{r_i}$ (from the reference frame r_i of template tracker to its current position) for all trackers,

where $i = 1..N_{tt}$ as the number of trackers:

$$ {}^{w}\mathbf{T}_r = {}^{r}\mathbf{M}_o^{-1}{}^{w}\mathbf{T}_o, {}^{w}\mathbf{T}_r = \{{}^{w}\mathbf{R}_r, {}^{w}\mathbf{t}_r\} \tag{8.5} $$

$$ \mathbf{H}^i = {}^{w}\mathbf{R}_{r_i} + \frac{{}^{w}\mathbf{t}_{r_i}}{d_i}\mathbf{n}_{r_i}{}^{\top} \tag{8.6} $$

Therefore we can propose a refiner for estimating camera pose and planar equation for multiple trackers homography. Note that we already know the correspondence mapping from $\{\mathbf{\Pi}^i\}$ to clustered and grouped planes $\{\mathbf{\Pi}^c\}$ by meanshift and data association, it means instead of regarding each plane separately from trackers, we set some planes in $\{\mathbf{\Pi}^i\}$ as the same during the optimization such that it has a sparse form:

$$ \{\widehat{\mathbf{q_w}}, \{\widehat{\mathbf{\Pi}_w^c}\}\} = \arg\min_{\mathbf{q_w}, \{\mathbf{\Pi}_w^c\}} \sum_{i}^{N_{tt}} \sum_{n}^{N_{p_i}} \left(\mathbf{p}_{w_i}^n - {}^{w}\mathbf{H}_{r_i}\mathbf{p}_{r_i}^n\right)^2 \tag{8.7} $$

with $\mathbf{p}_w^n$ and $\mathbf{p}_{r_i}^n$ are the corner points of template regions from current world frame and the corresponded reference frame of the template tracker i respectively, their sum quantity is denoted as N_{p_i}. Remember that the camera pose $\widehat{\mathbf{q_w}}$ and planar equation $\widehat{\mathbf{\Pi}_w^c}$ we are searching for are actually in world coordiante, thus a transform of Eq. 8.5 from global coordiante to local coordiante is mandatory as the homography is defined only between the reference frame and current one. For the reason of simplicity we abuse the term ${}^{w}\mathbf{H}_{r_i}$ and hide the transform in Eq. 8.7.

Warm starts for the parameters during the optimization can be given directly from last camera pose and also by searching for the previous global planar results for each template tracker. Unlike the previous work [130], with the help of template tracker, plane data association is no longer a problem as we already know that for each plane is generated by which template tracker. A simple searching and comparing of trackers will ensure the data association.

8.6.2 Bundle Adjustment-like Refiner

The Plane map refiner consists in an optimization framework which refines all keyframes' poses and their common planes found by plane matching process. Each keyframe contains multiple planes and keypoints in each plane. Once the *joint plane* information is gained over different keyframes, like global BA for point-based SLAMs, this procedure eliminates the drifting problem, solves scale ambiguity and refines camera trajectory w.r.t

whole sequence.

Analogically, we can develop a Bundle Adjustment (BA) system for refining every frame's pose and the *joint plane* information by minimizing mutually their re-projection error. Its functionality is similar to global BA for point-based SLAMs, it eliminates drifting problem, alleviates scale ambiguity and estimates more precisely camera trajectory.

$$\arg\min_{\mathbf{q}_t,\{\mathbf{\Pi}_t^c\}} \sum_{t}^{N_t}\sum_{i}^{N_{tt}}\sum_{n}^{N_{p_i}} \left(\mathbf{p}_t^n - {}^t\mathbf{H}_{r_i}\mathbf{p}_{r_i}^n\right)^2 \tag{8.8}$$

where t and i are the index of frame and tracker number, N_t and N_{tt} represent the total frame and template trackers number respectively.

8.6.3 Planar Map

Plane merging and keyframe

We also deploy a plane merging scheme to fuse close planes together given a metric on normal vector $\mathbf{n}$ and orthogonal distance d. Ideally we don't rely on well selected keyframes such as [130], as keypoint homographies usually meets difficulties if no enough translational baseline presents between two images. Template trackers allow us to track the plane on sequence and wait until the estimation is stable for proceeding computation of pose. Though we finally manually defined keyframes on certain image intervals to ease the computational cost of global BA optimization.

Template rejection

One disadvantage of template tracker lies on this robustness against outliers, unlike RANSAC-based methods, one outlier of template tracker is capable of polluting the result. Though we apply robust loss function such as Huber and Cauchy, it's still very essential to remove bad template trackers before they import too much noise into the system. Three main points are chosen here to filter out bad trackers:

— The unconvergence led by tracker's optimization, it usually happens when initializing on texture-less or non-planar regions.

— The voting distance during the ambiguity elimination process: if non of found solution is close to the common voted translational direction.

— Unstable templates: we monitor each template in terms of their plane equations. We prune trackers which fail to generate stably the plane equation in distance of

plane parameters.

8.7 Experiments and Discussions

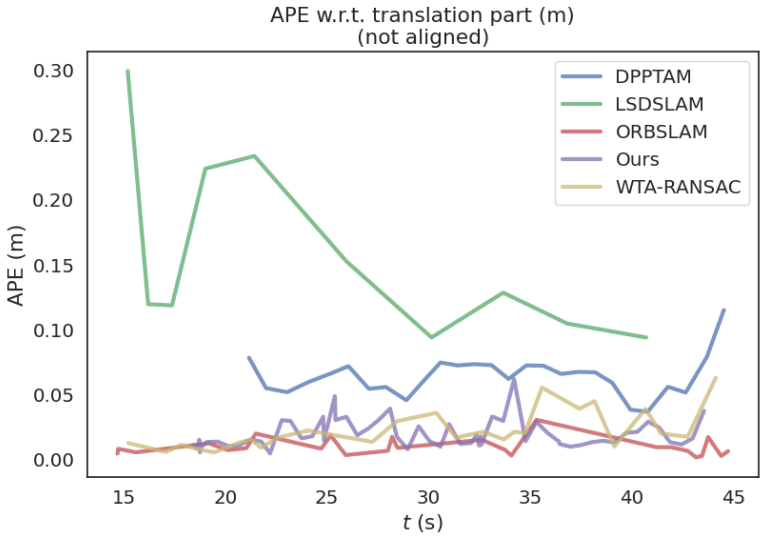


Figure 8.5 – Aboslute Pose Error (APE) metric for the sequence `fr_str_far` of Dataset [113], it demonstrates proposed method outperforms all dense and semi-dense methods and reach good precision level to ORBSLAM which only provides sparse point cloud map.

We test our proposed method in following two main different scenarios: indoor and outdoor environments:

For the indoor environment, we test three different levels of difficulty and complexity: single plane scenario, multiple planes scenario, and finally complexe multiple planar real room. Single plane scenario and multiple plane scenario are tested with the TUM RGB-D dataset [113] which is also tested by many planar or dense SLAM methods [23, 48, 130]. The scene is composed of rich textured planar structures and relative homogeneous color distribution on the wall area for multiple planar scene. It gives challengs for superpixel and template trackers as sometimes superpxiel might be spawned at the middle line of two different planes and mislead the following estimations. However the proposed system handles well the single multiple planar scene, see Table I for the comparison of Aboslute Pose Error with ORB-SLAM [77], LSD-SLAM [34], Multi-Level Mapping [48], DPPTAM [23] and relative pose estimation by WTA-RANSAC via superpixels [130]. Our method outperforms all semi-dense and dense methods and reaches a good level of precision against a

state-of-the-art monocular sparse keypoint-based SLAM [77] which only provides sparse point cloud mapping.

Comparison of Aboslute Pose Error (APE) for the sequence `fr_str_far` is demonstrated in Fig. 8.5 it is seen that the proposed method yields low level of error along the trajectory.

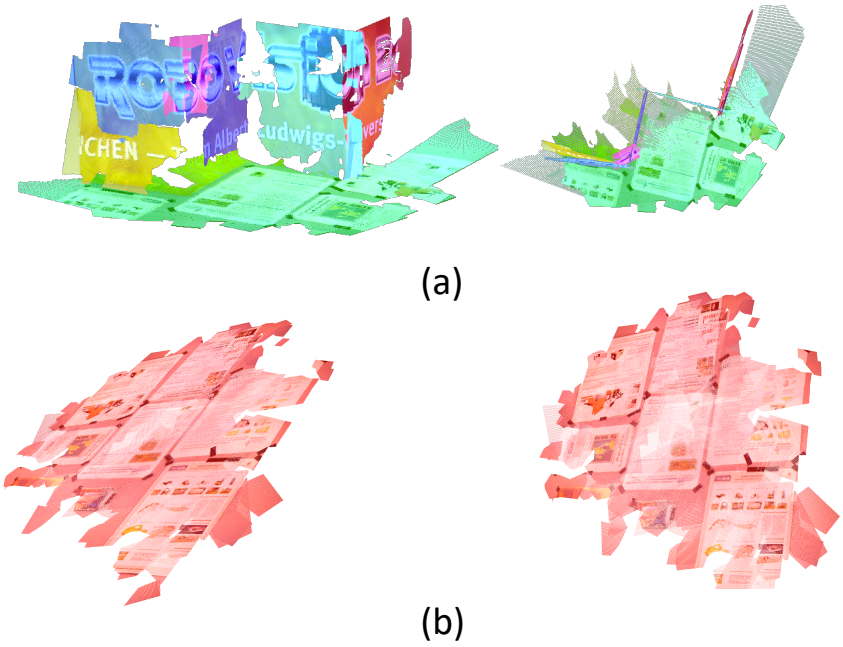


Figure 8.6 – 3D multiple (subfig a) and single plane map (subfig b) of the dataset TUM [113] generated by our method. Our proposed method is able to estimate camera trajectory and planar map representation simultaneously.

Single and multiple planar map and are viewed in Fig. 8.6, dense planar map is generated by reprojecting template trackers region according to found planar equations at each frame. It's observed that the map conserves well the perpendicularity without applying any Manhattan assumptions (*i.e.* forcing planes be perpendicular). One explaination about single planar map's precision drop is that without using keypoints and specially designed relocalization module, the system tends to accumulate errors along the tracking and negatively influenced by motion blur taken during the image acquisition.

Second experiment about indoor scene is deploied with Drone Dataset EuRoc[15]: a drone camera recorded graylevel dataset in a indoor test room. We take a segment of the long video ($\sim$ 400 frames) as the environment is not specific designed for planar SLAM and some textureless section and regions fails template trackers. In this experiment,

Data	*Methods*	*Mean (m)*	*Median (m)*	*RMSE (m)*
f3_str_far	[77]	**0.010**	**0.009**	**0.012**
	[34]	0.157	0.124	0.170
	[48]	-	-	0.17
	[23]	0.063	0.063	0.065
	[130]	0.023	0.017	0.027
	Ours	**0.018**	**0.014**	**0.021**
f3_str_far	[77]	**0.012**	**0.011**	**0.013**
	[34]	0.733	0.649	0.867
	[48]	-	-	0.22
	[23]	0.180*	0.159*	0.197*
	Ours	**0.110**	**0.098**	**0.120**
v1_1_e	[77]	0.091	0.085	0.094
	[34]	1.205	1.107	1.406
	[23]	x	x	x
	Ours	**0.099**	**0.080**	**0.112**

Table 8.1 – Evaluation of ATE of datasets. The proposed method outperforms DPP-TAM, LSD-SLAM and Multi-Level Mapping. Despite behind ORB-SLAM performance (a keypoint-based sparse SLAM technique without planar assumption), our approach provides a dense map representation (***** means lost a portion during tracking, **-** means no reported data, **x** means can not properply initialized).

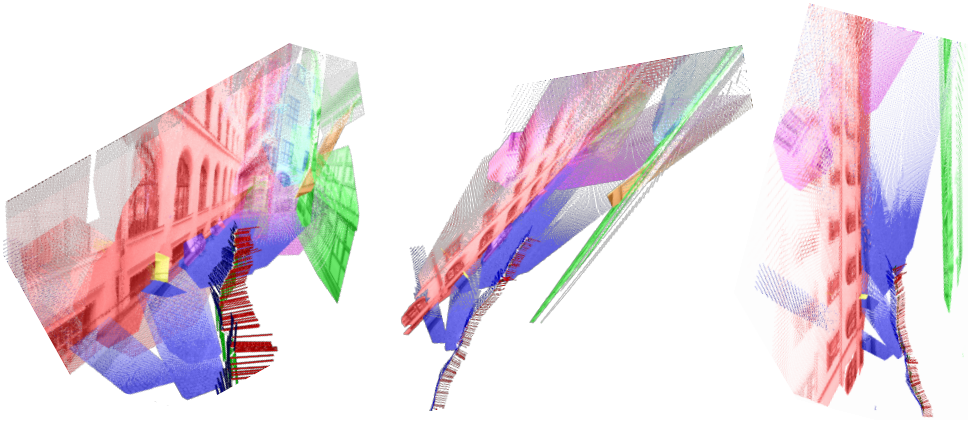


Figure 8.7 – Reconstructing on the dataset [35], coordinates represent the camera poses. The multi-planar environment is well conserved without applying Manhattan assumptions.

we also achieved a good level of precision than all dense methods and even better than ORB-SLAM on median error metric.

For the outdoor experiment, we test our system on image sequence from hand-held gray-level monocular dataset [35], in a scene of a corridor-like environment. Fig 8.7 displays that our system successfully recovers the corridor's planar structure as well as a camera trajectory from the input sequence. [1]

8.8 Conclusion

We proposed a novel way of estimating camera pose and generating a dense planar map with help of template trackers. Template spawning problem is solved via superpixelized image regions. A meanshift clustering on decomposed planes from homographies is applied for preparing data association and merging similar planes. Finally, we designate an optimization framework on corners of tracking regions for achieving better performance.

1. link to video: https://youtu.be/uBhOJc4HWq0

Chapter 9

Binary Graph Descriptor for Robust Relocalization on Heterogeneous Data

In this chapter, we propose a novel binary graph descriptor to improve loop detection for visual SLAM systems. Our contribution is fourfold: i) a graph embedding technique for generating binary descriptors which conserve both spatial and histogram frequential information extracted from images; ii) a generic mean of combining multiple layers of heterogeneous information into the proposed binary graph (BiG) descriptor, coupled with a matching and geometric checking method; iii) an implementation of our descriptor into an incremental Bag-of-Words (iBoW) structure in order to improve efficiency and scalability; and iv) a method to convert DNN (Deep Neural Network) results to our descriptor to improve robustness. We evaluate our system under synthetic and real datasets across different weather and seasonal conditions. The proposed method outperforms state-of-the-art loop detection frameworks in terms of relocalization precision and computational performance and displays high robustness against cross-condition datasets.

9.1 Problem Description

Loop detection is a key module of modern SLAM pipelines. Its rationale is to eliminate the drifting problem during the SLAM process by retrieving already seen locations and taking them into account during the Bundle Adjustment procedure. Two characteristics are crucial in the SLAM loop detection module: i) the capacity to retrieve already seen locations with high precision despite different views, lighting conditions and weather changes, and ii) the fast computing performance, which includes the detection of features from the current state and the query time within the database, which tends to increase rapidly as the sequence becomes longer. Widespread loop closures techniques [26, 44] classically rely on appearance-driven image retrieval methods with the help of compressed information or low-level handcrafted features for their low weight, fast computation, and good compatibility with the feature-based SLAMs [77, 89], such as Bag-of-Words (BoW) [110]. The technique of BoW is therefore widely applied in many visual SLAM (vSLAM) frameworks for loop detection tasks [77, 34, 89].

From another angle, with the rapid development of Deep Neural Networks (DNN), numerous works demonstrate robust performances on image retrieval and visual localization tasks, especially under extreme challenging conditions (*eg.* NetVlad [4], DenseVlad [120]). However, besides the GPU requirement, two main factors hinder a general application of network descriptors into SLAM algorithms: i) facing the increasing scalability and fast query demand for the SLAM loop detection, many networks provide only exhaustive searching methods on the generated descriptors (while classic BoWs rely an inverted indexing scheme to accelerate queries); ii) though many neural networks present high generality towards different input formats, it is complicated for supporting heterogeneous image layers (*eg.* depth, semantics, lidar point clouds) from differently designed networks. Specialization on network structure and training process seems inevitable.

This chapter proposes a generic Binary Graph (BiG) descriptor generated by a specific graph embedding technique on image regions. It improves loop detection with the heterogeneous image layers, including DNN outputs (Fig. 9.1). The motivation in using a graph structure lies in its generic and spatial-aware representation for supporting various inputs and outputting binary descriptors for a BoW framework. Thus, the proposed method benefits from both sides: the image discrimination capacity of heterogeneous information and the accelerated query process of the BoW design.

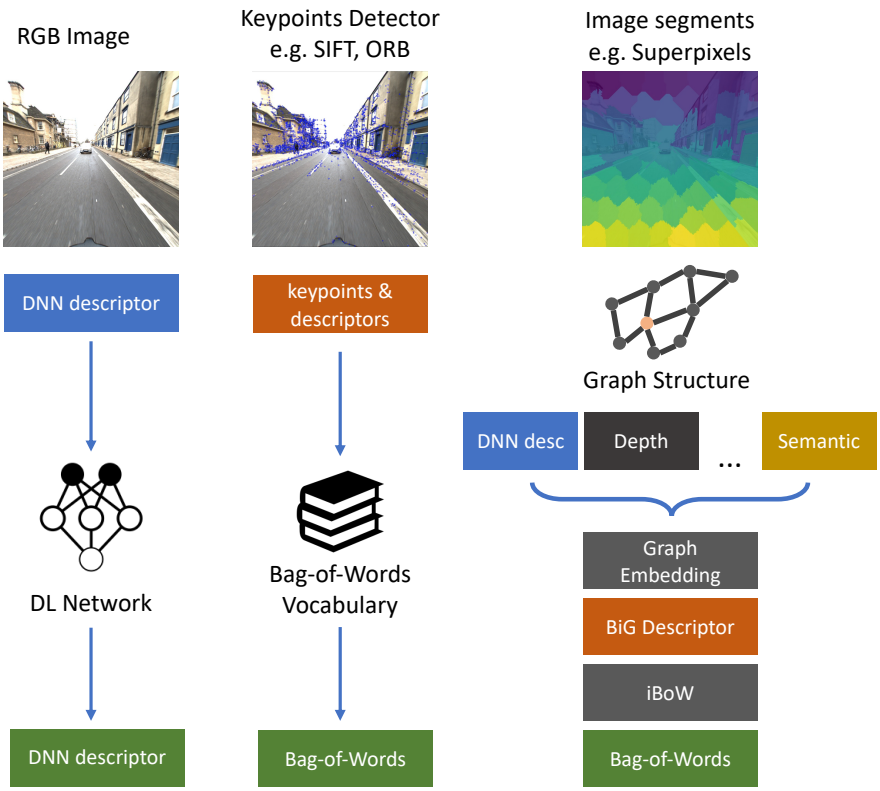


Figure 9.1 – We propose a generic graph binary descriptor on multiple heterogeneous image layers which supports iBoW for robust and efficient loop detection in SLAM applications.

9.2 Specific Related Work

Loop Detection Methods Mainstream loop closure methods can be divided into two categories: feature-based and image-based. The former group utilizes hand-crafted or learned local features for conducting loop detection (*eg.* : SIFT [68], ORB [100] and D2-Net [32]), whereas the latter group works on a global approach by exploiting the whole or patched image information. Feature-based methods are widely employed in the feature-based SLAM and visual-inertial odometry (VIO) systems [77, 34, 89] for its low computational cost good and compatibility with the extracted local features when estimating camera poses. Another advantage of the local features lies in its robustness towards perspective warps, illuminative changes and dynamic environments [129, 137]. Following this intuition, a series of local feature supported Bag-of-Words methods such as: FabMap [25, 26] and DBoW [44] are proposed and present in various SLAM implementations [77, 89, 34]. Both methods are based on the BoW principal for addressing

the matching problem. Importantly, the usage of *inverted indexing* technique reduces the computational cost of the query process against the increasing image database whereas the naive image retrieval methods iterate every image and yield cubic time complexity. The idea of the inverted indexing consists in storing the image index with the extracted word index to augment the filtering efficiency during the searching stage. Later, Garcia-Fidalgo and Ortiz [45] designed an incremental Bag-of-Words (iBoW) structure iBoW-lcd which, instead of training an offline vocabulary, can create an online vocabulary incrementally when the new features are added into the database.

Visual Localization and Image Retrieval: Entering the new era of Deep Neural Network (DNN), countless networks are proposed to handle the image retrieval task with the might of neural network. These networks achieved astonishing results compared to the traditional ones (*eg.* NetVlad [4], DenseVlad [120]). A brief intuition of these networks consists in generating a descriptor of a given image which conserves a robust metric against environmental changes and perspective warps. However, in the SLAM context, except the requirement of GPU computation, another predicament lies in its naive yet heavily searching scheme influenced by the scale of the database. Another branch also investigates the topic of visual localization but under a 2D-3D scenario. The problem is usually described as re-finding the local descriptors from the query image in an already generated point cloud with reference images. Some common interests between image retrieval and 2D-3D localization are shared as several works propose to rely upon the image retrieval method as a pre-processing step to narrow down the searching space for the 2D-3D localization algorithms [101, 118].

Semantic Segmentation and Graph Embedding: Semantic segmentation information is defined as a partitioning towards multiple segments containing different semantic labels according to human understanding and annotations. Besides local features, global hand-crafted and learned image features, some recent works demonstrate that semantic segmentation can help the loop detection too. [47] and [67] exploit random walk graph embedding technique [85] on semantic segmentation images for achieving seasonal and viewpoint robust loop detection results under indoor and outdoor conditions. Larsson et al. proposed a cross-season robust semantic segmentation [64] and a fine-grained self-supervised segmentation network FGSN [65] both demonstrate good performance in 2D-3D visual localization tasks. The latter work FGSN proposes a group of self-supervised segmentation classes robust to viewpoint, illumination and lighting changes instead of human annotated labels.

9.3 Proposed Methods

9.3.1 Overview

The proposed binary graph (BiG) descriptor balances well between the local feature and global spatial information, aims at improving the SLAM loop detection on heterogeneous image layers and benefits faster query process. Precisely, we list our contribution as follows:

— a generic binary graph embedding technique for computing descriptors while conserving spatial relationship and considering textual information from heterogeneous formats by virtue of graph structure

— a mean of combining multiple layers of information into our descriptor for improving the matching performance and robustness, coupled with a specific matching method and rough geometric checking scheme

— an implementation of our multiple layer descriptor into an incremental Bag-of-Words (iBoW) structure, for benefiting from the inverted indexing technique to accelerate the query process and increase the scalability when facing a fast growing database

— moreover, our descriptor shows a generic way to allow the use of the inverted indexing by DNN outputs with the help of our descriptor for fast loop detection tasks

The pipeline to generate the BiG descriptor (see Fig. 9.2) is composed of the following modules, which we will detail in the subsequent sections: 1) graph structure generation; 2) deterministic graph embedding; 3) histogram generation and binarization; 4) support of heterogeneous image layers.

9.3.2 Graph Generation

The rationale in building an image descriptor through graph embedding is that the graph structure preserves relevant characteristics for loop closure tasks. Unlike keypoints frequency-driven BoW methods which ignores the spatial information, or the global methods (*eg.* GIST [31], CNN-based global descriptor for localization [79], etc.) prone to be influenced by image distortion and noise, the description of images by graph representations covers both local and global levels of information. The representation of image regions considers pixel-level content while the graph structure provides a comprehensive

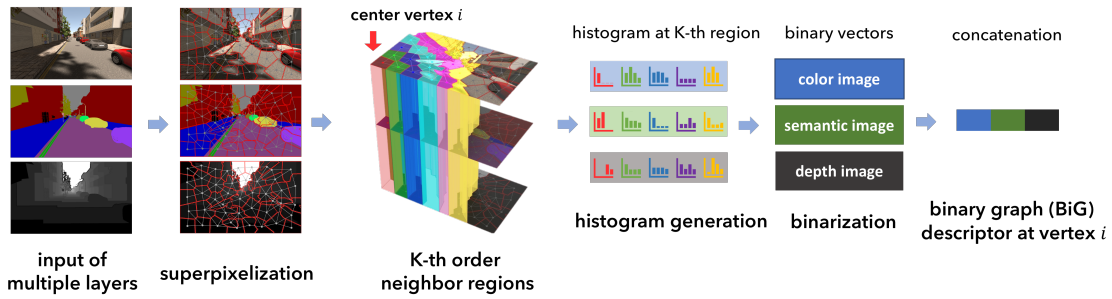


Figure 9.2 – The proposed pipeline to generate a BiG descriptor for a given superpixel (center vertex i). Multiple layers of heterogeneous image are structured in a graph representation spanning over neighboring regions of the superpixel, and embedding in a BiG descriptor through graph embedding and binarization of multiple layered histograms.

description of the image's spatial relations. Notably, the generality of graph structure representations enables a generic and extensible encoding of image characteristics which can encompass many different layers (depth, color, semantic, etc).

In this chapter, our graph structure is generated from connected superpixelized regions (*eg.* SLIC [1]). An unidirectional unweighted graph is: $\mathbf{G} = (V, E)$ where V the vertices are in a set of regions in image I, and E presents their adjacency (we build an edge when two superpixel regions are adjacent in an 8-connected layout). One advantage to use superpixel or other similar methods (*eg.* semantic or object segmentation) stands in its repeatability through different images: the generated regions are less sensitive to scaling and slight wrapping distortion since the region's color differences are preserved at a certain level under different views.

9.3.3 Neighbour Regions and Spatial Encoding

Different to Random Walk based stochastic graph embedding methods [85], we seek to design a deterministic graph embedding method: instead of visiting vertices in an arbitrary order, a deterministic method should output the same results when inputting the same graph and starting vertex.

Our proposed graph embedding method relies on the identification of neighbour vertices and regions. We define the k-th order neighbour vertices $N^k(V_c)$ of a given center vertex V_c as a set of vertices $\{V_i\}$ which have geodesic distance equal to k towards the center vertex V_c:

$$N^k(V_c) = \{V_i \in \{V\} : d(V_i, V_c) = k) \tag{9.1}$$

these neighbour vertices can be naively computed through a BFS (Breath-First-Search) search on the graph.

Once the definition of neighbour vertices is given, for a given center vertex, one is able to divide a graph into several united regions, in which each has a different neighbourhood distance $N^k(V_c)$ respectively. We term these order-related regions as $\Omega^k(V_c)$ (see Fig. 9.3).

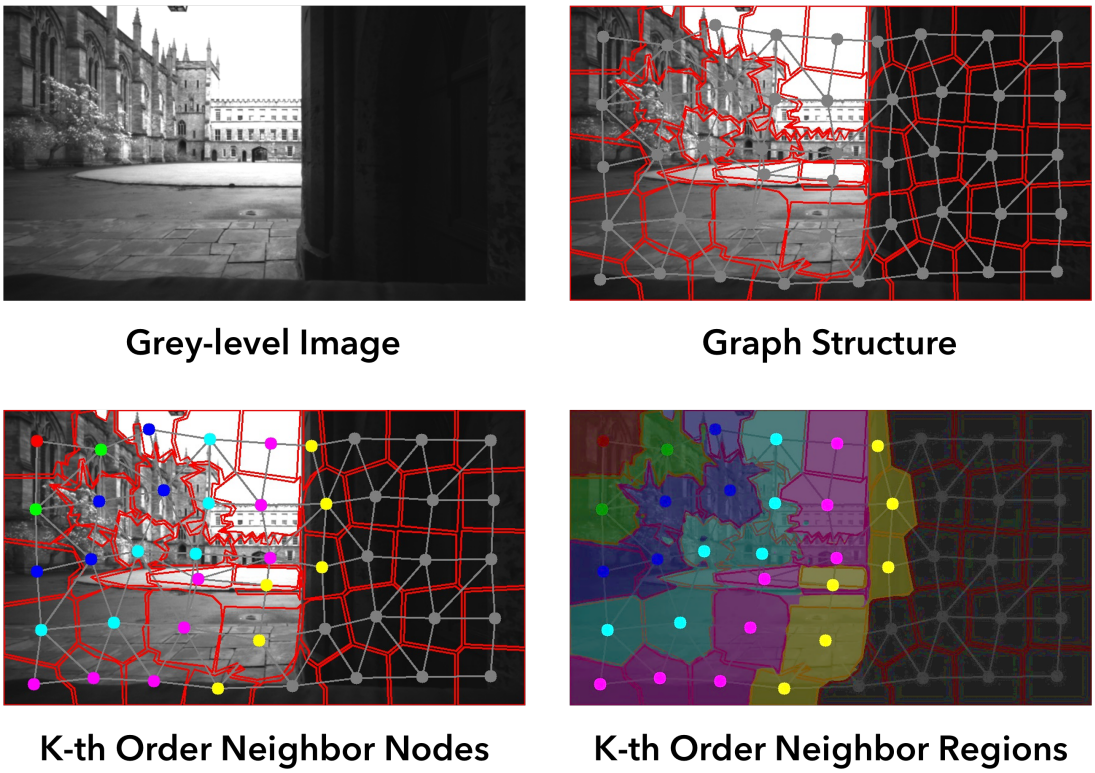


Figure 9.3 – A graph structure of superpixel connected regions is generated from an input image (left upper), red contour and gray nodes and vertices (right upper); given the center vertex of the red superpixel at left upper corner, we find k-th order neighbour vertices with different colors (red for $k = 0$, green for $k = 1$, etc.) (left lower) ; k-th order neighbour regions are highlighted in the image (right lower). Image from the Newer College Dataset [92].

9.3.4 Histogram Generation

Image histogram, frequency statistics of a image, is prevalent in many robotics vision applications such as: image retrieval [18], visual servoing [11], and image registration [29] as it presents good characteristics on describing and manipulating image information for tasks such as matching, tracking and registering, etc.

In this paper, we also rely on histogram for depicting image information and retrieving similar images. Different to common applications computing histogram in whole image space, we rely on regional histograms to compute *localized neighboring histograms* which exploits the graph structure to express both spatial and frequential information.

Describing a histogram as a real vector of a given bins number N_b: $h \in \mathbb{R}^{N_b}$, we calculate localized neighboring histograms in different k-th order neighbour regions $\Omega^k(V_c)$. Given a center vertex V_c, and its k-th order neighbour regions, a set of k histograms $h^k(V_c)$ (one per region) can be obtained by computing independently within each regional area expressed as $H^k(V_c)$:

$$H^k(V_c) = \{h^0(V_c), ..., h^k(V_c)\} \tag{9.2}$$

9.3.5 Deterministic Graph Embedding and Binarization

We then propose a deterministic graph embedding technique (nodes are embedded in a deterministic order in contrast with random-walk embedding techniques). Given our k-th order neighbour region histogram in Eq. 9.2, we access graph information by extracting pairs of histograms in a combinatorial order: given K neighbors in total, we select combinations of C_K^2 and sort them in ascending order: $(0, 1), ..., (0, K), (1, 2), ..., (K\text{-}1, K)$. The intuition of this operation consists in improving the discriminating capacity by computing a statistical difference and encoding spatial relationship into a binary vector in a deterministic order. Once the pairs of histograms is selected, we perform a binarization by comparing selected histograms w.r.t each bins b, similar to the BRIEF [17] operator:

For each bin b the comparison is simply:

$$\tau(h^i, h^j, b) := \begin{cases} 1 & \text{if } h^i(b) < h^j(b) \\ 0 & \text{otherwise} \end{cases} \tag{9.3}$$

We rewrite it into a more compact way for binarizing a pair of histograms of a given center vertex V_c:

$$\beta_i^j(V_c) = \tau(h^i(V_c), h^j(V_c)) \tag{9.4}$$

where $h^i(V_c)$ is i-th order regions histogram, same for j.

Finally, we generate a long binary sequence vector (*i.e.* descriptor) B_c via concatenat-

ing the binarized histograms together in combinatorial order.

$$B_c = \beta_0^1(V_c) \oplus, ..., \oplus \beta_{K-1}^K(V_c) \tag{9.5}$$

where the $\oplus$ is concatenation operation, the length of our binary descriptor B_c is $N_B = N_b \times C_K^2$. *eg.* , if we use histogram of 32-bins and 5 orders for neighbor regions, B_c has a size of $32 \times C_5^2 = 320$ digits for each vertex in graph.

Given a vertex V_c, we can compute a B_c for describing its region and the relationship towards its vicinity. A series of binary descriptors $\{B_c\}$ can therefore be generated for a whole graph (*i.e.* input image).

9.3.6 Generic Multiple Layer Descriptor

Applying the deterministic graph embedding has another advantage: ch binary digit's order is determined and corresponds to its combination order given a graph structure, *i.e.* its spatial relation. Therefore, we can easily concatenate the binary sequence descriptor with other layers of information as long as they share an identical graph structure.

For example, under the scenario of RGB-D camera, we can apply the previous process for extracting a descriptor B_c^g with a given center vertex V_c on a color image. Similarly, one can also compute the histogram of a depth image (after transforming the images from two sensors in a same coordinate) with the same graph regional segmentation and generate a descriptor B_c^d at the exact center vertex V_c. A concatenation operation helps to combine and generate an extended descriptor B_c^* for this multiple layer heterogeneous information (see Fig. 9.2).

$$B_c^* = B_c^g \oplus B_c^d \tag{9.6}$$

The advantage of this design lies in its compact structure and low cost for appending more layers. This provides a simple and fast way to fuse the heterogeneous data from different sources such as depth image, lidar information, semantic images, even the DNN results, though each type of data requires specific pre-processing before integration in our representation.

Depth Image: For depth images acquired by depth sensors, we treat them like grey-level images to generate descriptors.

Lidar Information: Lidar (Light Detection and Ranging) sensors yield 3D point cloud rather than 2D images. Related work believe its difficulty consists in its sparse and three-

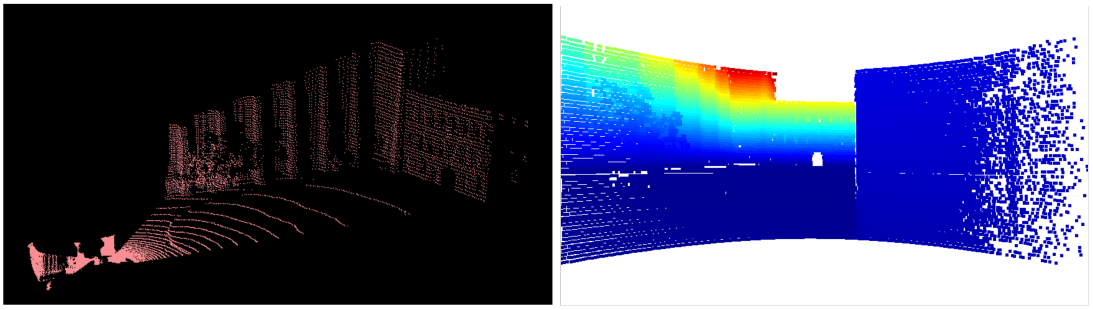


Figure 9.4 – We project point cloud data into 2D depth images along the camera frame coordinate. The blank arc regions are caused by the limited field-of-view of Lidar device.

dimensional nature. In this chatper, we render the point cloud to images as the extrinsic calibration between the camera and Lidar sensor is known (see Fig. 9.4). Though the sparse nature of Lidar data raises problems for local descriptors, histograms remain an appropriate tool that ignores precise spatial information and yields statistical features. In other words, once we project point cloud to images, one can treat them as grey-level images for fitting into the system.

Traditional Semantic Image: For semantic images, the histogram shall not be built on pixels intensities but on labels. Therefore the length of the histogram is not arbitrary, as the quantity of labels is given and fixed. Following the idea of [47] and related work, we manually divide labels into two categories: robust and unrobust. Robust labels include buildings, plantations, etc, the unrobust ones are mostly non-permanent objects, such as pedestrians and vehicles.

FGSN Semantic Segmentations: FGSN [65] is a self-supervised semantic segmentation method which doesn't provide precisely the meaning for each learned label. Labels are learned for conserving high robustness against environmental variations. We treat FGSN as semantic types but without robust/unrobust separation.

Neural Network Results: Many network driven image retrieval methods rely on environmental invariant image segmentations such as the aforementioned FGSN [65], whereas the mainstream methods usually output directly descriptors of the whole image: *eg.* NetVlad [4], DenseVlad [120]. For fitting DNN results into our system, we generate respectively a descriptor for each region, and vectorize them directly as histogram.

9.3.7 Matching Descriptors and Geometric Checking

During the retrieval stage, namely the matching process, we need to compare and score two sets of descriptor vectors from the candidate and the query images respectively. Commonly, the simplest way is to rely on the Brute-Force Hamming distance matching approach, then average and normalize the computed Hamming distances on the matched descriptors to indicate the similarity between two images. However, unlike other well-defined metrics, a Hamming distance can rarely touch zero (it means inverted XOR digits), therefore causing an insufficient discrimination ability.

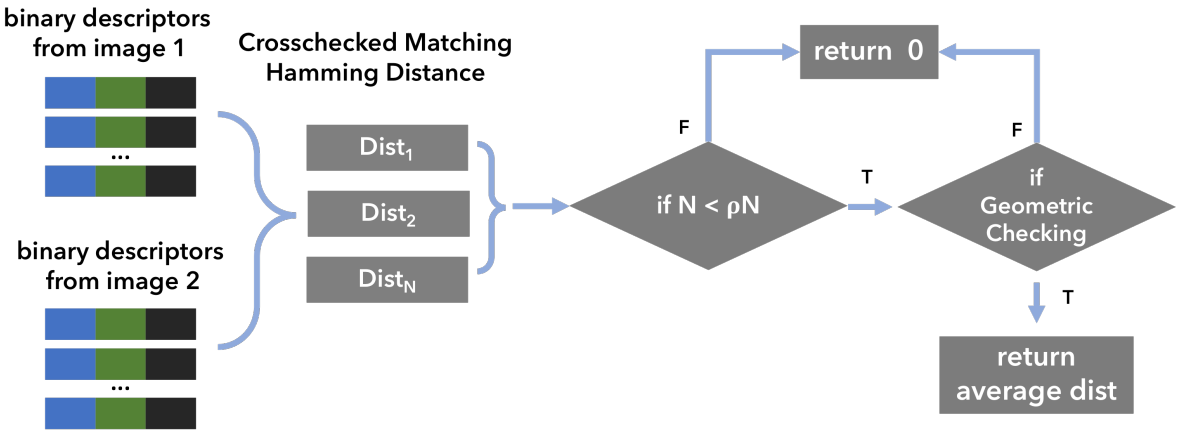


Figure 9.5 – A pipeline of the matching process of two descriptors. The result is controlled by two stages: it outputs zero (most dissimilar) when low matching number than ρN after a crosschecked matching or high geometric checking error.

To fix this problem, we set a threshold ρ on the matching number to remove candidates of insufficient matches, as this often suggests multiple wrong matches colliding with each other. Another improvement is proposed at the geometric checking stage. Similar to keypoint BoW methods using fundamental matrix to compute and remove unqualified candidates, we apply affine constraint on each region center position between the query and the found candidates, and use the reprojection error for thresholding candidates since geometrically the image regions' positions of the same place should roughly satisfy the constraint. It adds the extra cost of computing RANSAC but yields better recall performance. We set two thresholds to improve the discrimination ability by outputting 0 if the conditions are not satisfied. Finally a normalised similarity is generated between $[0, 1]$; the matching process is described in Fig. 9.5.

9.3.8 Loop Detection System and Incremental Bag-of-Words

In the SLAM context, loop detection tasks are generally reduced to mere image similarity measuring tasks. The main differences are twofold: i) unlike image retrieval tasks concentrating on the recall capacity [4], loop detection problems often emphasize the precision over recall factor. The reason lies in the necessity of avoiding erroneous matches that would collapse the optimization; ii) loop detection techniques need to maintain low computations times not only in generating descriptors and matching them, and also limit the increasing overall query time as the database of locations grows with the sequence. The latter factor is usually ignored under the context of the image retrieval problems.

To address the first factor, DBoW [44] and many following works propose to apply a series of matching rules on the sequential groups of keyframes, (refer to the [44] for more details) and improve the precision performance during the detection. On the other hand, BoW-based [26, 44] methods rely on a technique named *inverted indexing* which helps improve the query speed by storing the keyframe index into the words for a fast pre-pruning of the searching space.

Classic BoW structures require pre-trained offline vocabularies which need to be known beforehand. Yet our descriptors are generated on the fly and the representation of binary digits varies depending on the different combination of heterogeneous layers, which raises difficulty to implement easily into traditional BoW systems.

Recently, an incremental Bag-of-Words approach has been designed (iBoW-lcd [45]), which performs vocabulary clustering in a online incremental approach. Furthermore, the matching rules of DBoW [44] are also conserved and developed in the method. We therefore expressed our binary descriptor into the iBoW-lcd structure in order to achieve high precision and faster performance in visual SLAM tasks.

9.4 Experiments

9.4.1 Datasets and Methodology

To demonstrate the capacity of our BiG descriptor to handle heterogeneous image layers, we relied on multiple public datasets of different types: i) Synthia Dataset [97], a synthetic dataset with color images, depth images, and semantic labels across different seasons, time, and weather conditions *eg.* summer, fall, night rain, etc; ii) Newer College Dataset [92], a real recorded dataset providing infrared images acquired by a Realsense

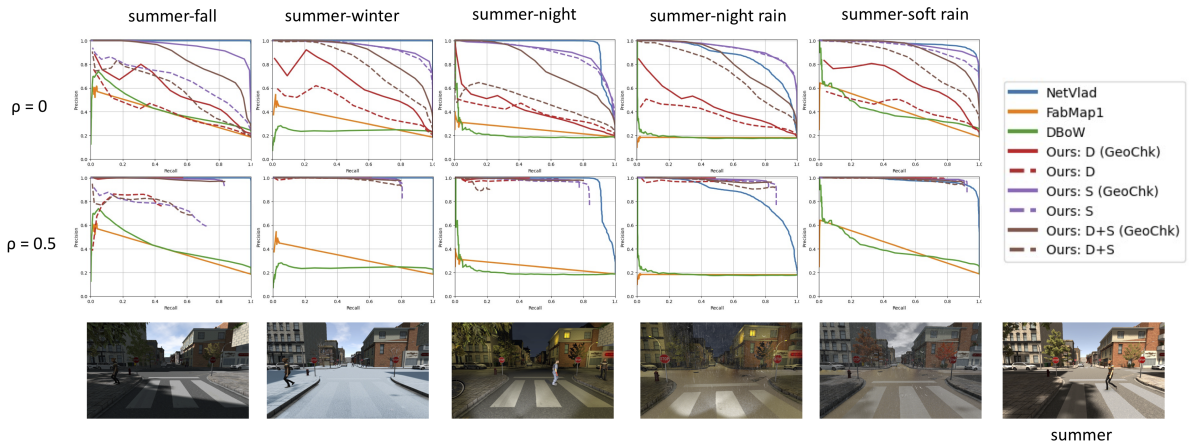


Figure 9.6 – The PR-Curves (x,y for recall and precision) demonstrate that our BiG descriptor outperforms DBoW and FabMap on all conditions by all layer combinations and show better results against NetVlad under rainy weather conditions. The improvement of ρ and geometric checking can also be observed in the comparison between first and second row, dot line and solid line. In the legend, D and S are Depth and Semantic layers respectively, D+S is the combination of two layers.

camera and Lidar point clouds on a relatively lower frequency; iii) RobotCar Seasons Dataset [102], a dataset across different conditions including night and rainy scenarios acquired by a car-mounted camera.

9.4.2 Synthetic Data: Across Different Seasons

We first test our proposed method on synthetic season change scenarios. Synthia [97] is a synthetic car-driving dataset on multiple scenes, including various seasonal changes, day-night shift and weather variations. Besides color images, the dataset also provides ground truth semantic labels and depth images and is widely used in relocalization tasks with heterogeneous types of data [47].

In this experiment, we focus on the cross environmental image retrieval capacity. The idea is to retrieve every image from one condition to another: *eg.* from the night and rainy condition towards a sunshine one. PR-curves are generated by tuning acceptance similarity threshold on metrics.

In the experiment, we test our proposed method on different combinations of image layers such as Depth and Semantics (shortly D+S in the results). An ablation experiment on matching number removal threshold ρ and geometric checking is also proposed. Comparison with related work includes i) DBoW [44]: a binary BoW method for SLAM loop

detection; ii) FabMap [26] a probabilistic BoW method and iii) NetVlad [4] a deep-learning method specifically designed for image retrieval under different conditions.

PR-Curve results are shown in Fig. 9.6. The proposed BiG descriptor handles well the image retrieval task under all conditions (with $\rho = 0.5$), specially the rain and night weather). Netvlad method demonstrates good results under daylight and non-rainy conditions but our method reaches higher recall on rainy conditions. The improvements of the matching number removal threshold ρ and the geometric checking in BiG descriptor are obvious through the comparison between the first and the second row and the one between dot line and solid line in the figure. Two layers combination (D+S) shows the best working point (highest precision and recall) on two night conditions after apply ρ and geometric checking.

9.4.3 Newer College Dataset: under Static Environment

We then test our method under a static environment scenario, similar to traditional use-cases in loop detection systems. We took the Newer College dataset [92] for also demonstrating the compatibility of Lidar data with the BiG descriptor. The Newer College dataset provides stereo infrared images with point clouds acquired by the Lidar device in an asynchronised approach. After manually synchronising the point cloud data with infrared data, we extracted 505 images with maximum distance at least 300 meters. We apply a distance threshold of 25 meters and an orientation threshold of 25 degrees for generating all ground truth loop closure event frames. Five methods are compared: our method implemented into the iBoW-lcd system, iBoW-lcd, DBoW and FabMap (1 and 2 on differently trained vocabularies).

Following the experiments of DBoW [44], we set the neighbor ground truth loop closure event as recalled when a true positive is found close enough (2 frames). This avoids a too low recall caused by the inter-competing of the candidates (*eg.* an outstanding candidate may compete and attract the true recalls of its neighbors lead to lower recall).

We project the lidar point clouds into depth images to adapt to the BiG descriptor according to the extrinsic calibration results from the dataset (see Fig. 9.4). Due to the limited Lidar field-of-view, some undetected regions exist on the depth image. We tested combinations including infrared image (IR), Lidar data, NetVLAD [4], and two semantic data (generated by DeepLabv3+ [21] and FGSN [65] a learned semantic label) with different pre-processing stages proposed in Section 9.3.6.

For the DBoW [44], three geometric checking configurations are evaluated: without

Methods	Precision	Recall
FabMap 1 (11k voc)	0.477	0.510
FabMap 2 (4k voc)	0.528	0.452
DBoW (NoGC)	0.636	0.164
DBoW (GCDI2)	0.636	0.164
DBoW (GCExhaustive)	1.000	0.317
iBoW-lcd (1k ORB)	1.000	0.522
Ours: IR + Lidar	1.000	0.298
Ours: IR + Semantic DeepLabv3+	1.000	0.471
Ours: IR + Semantic FGSN	**1.000**	**0.538**
Ours: IR + NetVLAD	**1.000**	**0.567**
Ours: FGSN + NetVLAD	**1.000**	**0.586**

Table 9.1 – Table of working point (Precision, Recall) for each methods, our proposed methods gain highest recall with the combination of infrared image and Netvlad DNN descriptor.

checking (NoGC), with a checking on node DI2 (GCDI2), and with an exhaustive checking (GCExhaustive). We tested FabMap1 [26] on an $11k$ vocabulary from the paper and train a $4k$ vocabulary on a similar dataset for the FabMap2 [25].

The results are shown as the best working point (highest precision point when the recall is non-zero) in the Table. 9.1, we observe that all configurations of our method can yield a working point with precision at 1.0 and relative high recall level. DBoW with geometric checking shows similar results over the combination of infra image and lidar projected depth image. ibow-lcd has the better results except our methods, however, by combining with Netvlad and semantic information, our methods outperform others.

9.4.4 RobotCar Season Dataset: under Changing Conditions

Finally, we demonstrate our method under changing conditions acquired by real cameras. RobotCar Seasons dataset is captured by car-mounted cameras across different seasons (summer, winter, etc), weathesr (rainy, sunny, snowing), and times (dawn, dusk and night). The reference dataset includes around $7k$ images taken under the overcast condition, and various query datasets have around 250 images each for different conditions. Given its challenging nature, many works exploit this dataset for robustness experiments [102, 65, 3]. In this experiment, we only use color images for image retrieval and loop detection purpose.

We perform two categories of experiments in this section: i) image retrieval experiment:

	day conditions							night conditions	
	sun	oc-summer	oc-winter	dusk	dawn	rain	snow	Night	NightRain
	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m	0.25/0.5/5 m
	2/5/10 deg	2/5/10 deg	2/5/10 deg	2/5/10 deg	2/5/10 deg	2/5/10 deg	2/5/10 deg	2/5/10 deg	2/5/10 deg
NetVLAD	2,87/12,44/85,17	0,04/25,91/95,91	2,97/19,31/86,63	13,37/37,97/90,37	3,48/17,39/80,43	7,58/25,25/93,43	6,69/24,27/90,38	0,51/2,03/15,23	0,89/3,57/24,55
NetVLAD (S)	3,35/15,31/85,65	5,91/35,45/97,27	8,42/35,64/89,11	18,18/43,85/92,51	7,39/27,83/83,48	10,10/32,32/93,43	10,88/28,87/91,63	0,51/1,52/16,75	0,89/4,46/21,43
NetVLAD (D2)	2,87/15,31/90,43	5,00/36,36/97,73	6,44/34,65/91,58	17,65/42,25/90,91	6,96/25,65/84,78	9,09/27,78/93,94	8,79/29,29/91,63	1,52/3,55/28,93	1,79/7,59/42,86
BiG FGSN	1,44/9,57/71,77	2,73/15,00/81,82	1,98/18,32/79,21	9,63/24,60/81,82	3,04/13,04/79,13	6,57/17,17/88,38	3,35/13,39/78,24	3,55/10,15/58,88	2,23/6,25/63,84
BiG F (S)	3,35/16,27/81,34	6,36/29,09/95,45	7,92/33,17/87,62	17,11/43,32/90,37	6,96/28,70/86,52	10,10/32,83/92,93	9,62/27,20/89,54	1,52/4,57/44,67	2,23/5,80/51,79
BiG F (D2)	5,26/17,70/89,47	4,55/31,36/95,45	6,44/38,12/92,08	16,58/40,11/90,37	6,52/27,83/87,39	10,10/29,29/92,93	7,95/28,03/91,21	3,55/10,66/76,14	3,57/13,84/86,16
BiG F+N	3,35/13,40/76,08	3,18/17,27/86,82	3,47/21,78/81,68	10,70/26,74/88,24	5,65/18,70/83,04	8,08/20,71/90,40	4,18/16,32/84,10	2,03/6,09/44,67	3,13/10,27/66,07
BiG F+N (S)	3,35/15,79/81,82	6,36/34,09/97,27	7,92/33,66/89,11	16,58/42,78/91,98	7,83/30,43/86,96	10,10/32,83/94,95	8,79/26,36/87,87	0,51/2,54/35,03	2,23/7,14/48,66
BiG F+N (D2)	4,31/19,14/90,43	5,00/29,55/96,82	6,93/35,64/92,57	16,58/41,18/91,44	7,83/29,57/87,39	10,10/28,79/93,94	7,11/28,03/89,12	1,02/7,11/74,62	4,46/17,41/84,82
Gain from NetVLAD	**1,44/3,83/0,00**	1,36/-2,27/-0,45	**1,93/-0,72/0,99**	-1,60/-1,07/-0,53	**0,87/3,91/2,61**	**1,01/5,05/1,01**	-0,84/-1,26/-0,42	**2,03/7,11/47,21**	**1,79/6,25/43,30**

Table 9.2 – Comparison of our method with one layer: FGSN semantic label (BiG F) and two layers: semantic plus Netvlad (BiG F+N) and original Netvlad on the RobotCar Dataset for different conditions. Three versions are applied on each method: i) best ranking candidate; ii) SIFT geometric checking on top 10 ranking candidates (S); iii) D2-Net geometric checking on top 10 ranking candidates (D2). The gains of the best results of ours against the best of Netvlad are in last row.

following the methodology of experiments in [65, 64, 102], we organize the experiment as follows: relying on the proposed method, we apply image retrieval experiment from the query conditions towards the reference condition for all images. The measure is defined as the percentage of images from the query dataset have been relocalized under certain distance and angle thresholds. The purpose of this experiment is to demonstrate the discriminative capacity of the proposed BiG descriptor with the combination of various layers of information such as FGSN [65] semantic labels and NetVald [4] DNN outputs.

Image Retrieval Experiment

In Table. 9.2, we test two types of combinations of layers: FGSN as single layer (termed BiG F in the table) and the FGSN plus NetVlad as two layers (termed BiG F+N in the table). With three versions tested for each combination: i) we directly take the first ranked candidate as the output of the query result; ii) we perform an extra geometric checking with the help of SIFT [68] (S in the table) keypoints within the top 10 ranked candidates and output the final candidate according to the geometric checking results; iii) the process is similar to the second one, instead of using SIFT keypoints, we choose a DL keypoint D2-Net [32] (D2 in the table) which demonstrates better robustness against environmental variations.

From the Table. 9.2 we achieve some observations: i) the geometric checking improves for all method: the D2-Net demonstrates better performance under difficult scenarios whereas the SIFT feature works well under less extreme conditions; ii) our method maintains good performance on daylight conditions, though the gain is slightly negative

		mean daylight	mean night
2D	Our method	**8.58 / 31.73 / 92.26**	**3.56 / 12.25 / 81.15**
	Netvlad [4]	5.29 / 23.22 / 88.90	0.70 / 2.80 /19.89
	Netvlad (GC)	7.98 / 30.66 / 91.80	1.65 / 5.57 / 35.90
	DenseVlad [120]	7.71 / 31.26 / 92.26	1.00 / 4.45 / 22.70
	FabMap [26]	2.80 / 12.34 / 30.37	0.00 / 0.00 / 0.00
	SeqSLAM [76]	1.30 / 6.10 / 15,30	0.20 / 0.70 / 1.50
	ToDayGAN [3]	-	2.15 / 11.00 / 50.20
3D	FGSN [65]	-	11.00 / 28.40 / 45.20
	DomainAdapt [8]	-	20.65 / 42.50 / 52.15

Table 9.3 – Comparison on average relocalization rate of our method with mainstream 2D image retrieval methods and 3D relocalization methods. GC refers to the geometric checking.

($< 1\%$) for few, the average result outperformed the geometric checking version NetVlad (See Table. 9.3); iii) the proposed method shows drastic improvements ($\sim$45%) on night conditions and proves the robustness and discrimative capacity of our descriptor; iv) the combination of two layers improves the performance during the daylight condition but lowers for the night ones, due to the bad performance of NetVlad during night (see Sec. 9.4.6 for more discussion).

In Table. 9.3, we list the average of the best performance of our methods with two combinations and two geometric checking methods, the Netvlad method with and without geometric checking and other 2D-2D (image retrieval) methods. The proposed method outperforms all the image retrieval methods, and especially gained drastic performance under night condition even compared to the specific designed DNN method for dark environment [3] and the state-of-the-art 2D-3D relocalization methods. In 2D-3D methods, the relocalization is achieved in an already built 3D map as a priori. Therefore the difficulty of the retrieval problem is eased and better results can be achieved numerically. Still our proposed 2D-2D method shows a gain of at least 30% even compared to the constraint relaxed 2D-3D methods in night conditions. More specifically, the FGSN [65] is exactly the semantic labels used for our proposed method as our method is 35% better on results when using the FGSN as the only layer. The improvements on identical input data shows the efficiency of our binary graph descriptor.

	sun	oc-summer	oc-winter	dusk	dawn	rain	snow	night	night-rain
BiG FGSN	**36.3/85.4**	36.8/97.6	50.5/97.1	58.3/90.1	**76.5/93.1**	83.8/96.5	39.8/94.1	**38.6/83.5**	**65.2/75.3**
BiG F+S	30.6/85.3	40.0/97.8	**58.9/98.4**	**78.1/95.4**	71.3/91.6	**88.9/99.4**	**56.6/99.1**	18.3/66.7	26.8/72.3
iBoW-lcd	8.6/89.5	**55.9/99.1**	53.9/99.1	75.4/94.0	33.9/89.7	87.9/99.3	40.6/100	0.00/0.00	0.00/0.00
DBoW	2.8/100	9.55/91.3	13.9/100	45.4/97.7	22.6/98.1	49.0/100	23.6/96.5	0.00/0.00	0.00/0.00

Table 9.4 – Comparison of Recall/Precision on different environmental conditions. we take 5.0 m and 10 deg as threshold for computing results, the best performances are highlighted as the maximum of precision plus recall.

Loop Closure Experiment

The second type of experiment aims at testing the loop closure ability of the proposed method by integrating BiG descriptor into an incremental bag-of-words loop detection framework. The motivation is to demonstrate the proposed descriptor can complete loop closure tasks under difficult scenarios with high precision, fast speed and good compatibility with all iBoW systems and the ability of combining heterogeneous information.

The methodology is similar to Sec. 9.4.3, we concatenate all images of reference condition and query images of various conditions respectively for inputting into a loop detection system sequentially. We mark the working point of highest precision and recall as final results. In this experiment we compare to the original iBoW-lcd [45] method which utilizes 1k ORB [100] as feature to build BoW and DBoW [44].

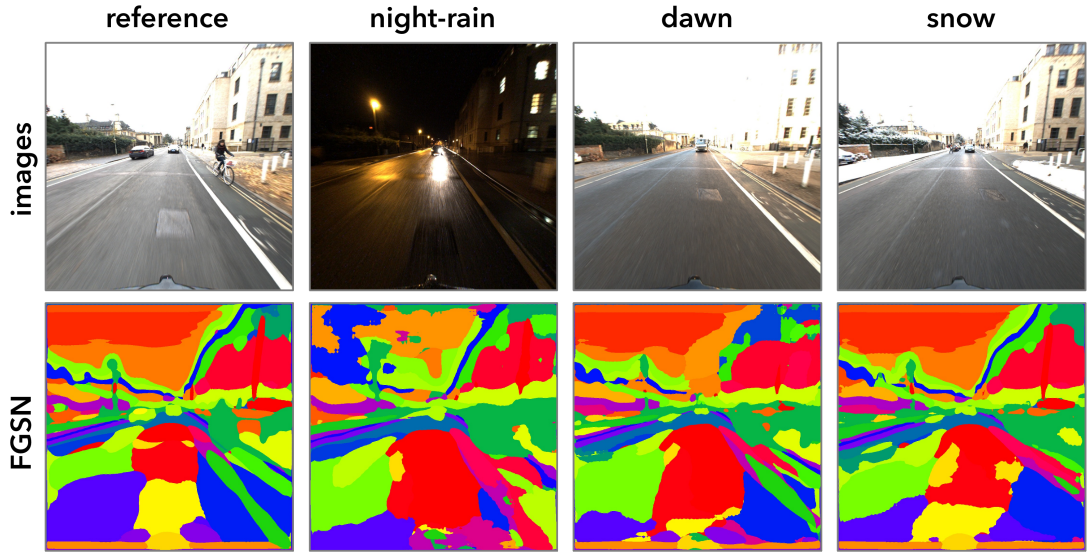


Figure 9.7 – The retrieval results from reference to various environmental conditions, our method can retrieve seen locations despite appearance disparities thanks to the spatial similarity in terms of semantic distribution on FGSN images.

After integrating the proposed BiG descriptor with multiple layers into an incremental BoW framework, the proposed method is able to outperform the original iBoW-lcd method in both daylight and night conditions with a higher recall and similar yet high precision (See Table. 9.4). The iBoW-lcd method generates better performance under very well lit oc-summer condition and lacks robustness against environmental variation such as sun, nights and dawn conditions.

9.4.5 Computational Efficiency

As we mentioned in Sec. 9.3.8, the binary descriptor displays a good compatibility with BoW techniques not only for improving the retrieval ability but also for achieving a faster query supported by the inverted indexing technique.

Methods	Ours	iBoW-lcd	FabMap	DBoW	Netvlad*	Densevlad*
Time (ms)	73.5	715.2	57.1	262.5	137	338

Table 9.5 – Comparison of average query time on the RobotCar Season Dataset per image. *: data from [102].

In the Table. 9.5, we measure our speed performance on a 2.7GHz Intel i7, as well as the iBoW-lcd, FabMap and DBoW. The other results are taken from [102]. We can see the proposed method only takes 73.5 ms whereas the other mainstream methods are more time consuming except FabMap. Netvlad and Densevald data are taken from [102] with a Intel Xeon E5 2.6GHz. The explanation for the difference between our method and iBoW-lcd lies in the different descriptor number: we only generate 50 descriptors for each image (one per region), but iBoW-lcd uses 1k ORB features. See Fig. 9.8 for the trend of the query time vs. increasing frame number: the query time of the iBoW method increases slowly with the inverted indexing technique, while the linear search method (traditional image retrieval) suffers the large scalability. Therefore the proposed BiG descriptor shows another advantage: converting the network outputs into binary descriptors for using inverted indexing accelerates the query time when facing large scale datasets. [1]

1. link to video: https://youtu.be/nQM1g83D85w

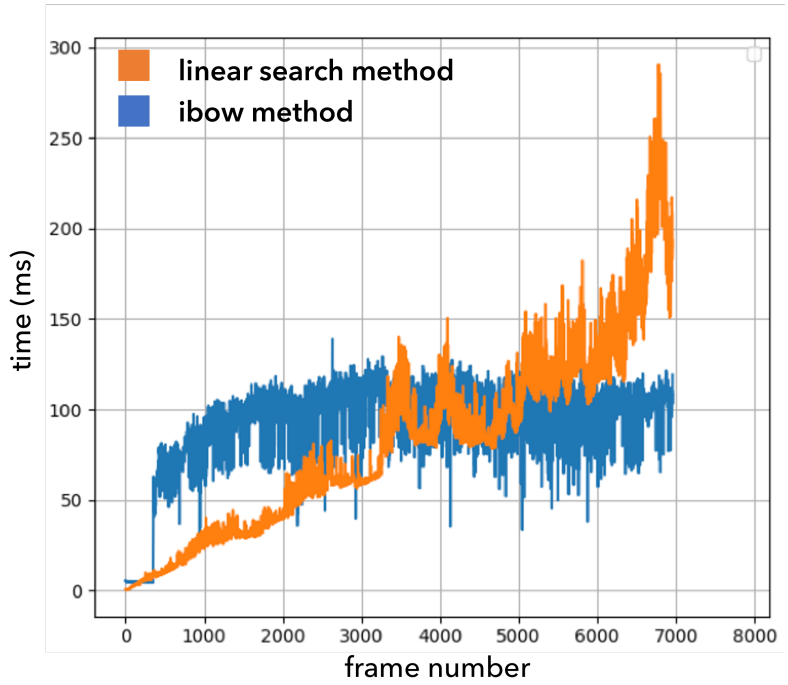


Figure 9.8 – Comparison between the iBoW method and linear search method used by common image retrievals. With inverted index technique, the iBoW methods shows constrained query time against the increasing scale, whereas the query time of linear search method grows proportionally.

9.4.6 Limitation

One limitation of our approach lies in the multiple layer combination strategy: concatenating the binary descriptors of multiple layers implicitly means an averaging of multiple inputs. Therefore, in the Table. 9.2, when appending an extra layer Netvlad with FGSN, the descriptor can not intelligently select more informative bins and causes lower performance in night conditions (BiG F+N). A possible future work lies in the smart control of this combination problem.

9.5 Conclusion

In conclusion, we proposed a binary graph (BiG) descriptor which is able to: i) encode image content and spatial information from a graph structure through the design of a graph embedding method; ii) combine heterogeneous layers of information such as semantic images or neural network results to gain better retrieval capacity under dynamic

environments; iii) rely on incremental Bag-of-Words structure to achieve higher precision and real-time performance for SLAM loop detection tasks; iv) provide a generic mean for DNN methods to exploit inverted indexing technique in BoW to accelerate query process against increasing database.

CONCLUSION

9.6 Conclusion and Perspective

This thesis targets from multiple angles and addresses the problem of robustness in visual SLAM systems: they can be categorized into three related modules local features for relocalization, mid-level features via keypoints and templates for tracking and mapping, and loop detection with a graph-aware binary descriptor. More specifically, in chapter 6, we proposed the concept and Mutual Information assisted computation of Multi-Layered Image (MLI) to improve the keypoint matching performance under difficult scenarios. In chapter 7 and 8, planar mid-level feature is exploited to propose multiple planar SLAM methods, through RANSAC-based and template tracker techniques respectively. Finally in Chapter 9, we discussed the loop detection problem and proposed a binary graph descriptor (BiG) which is compatible with heterogeneous data and Bag-of-Words (BoW) system, to help the robust SLAM loop detection under different time, weather and season conditions.

The perspective shall also comply with the main structure of the organization. It comprises multiple layers and tries to cover different angles:

From very technical angles:

— In the module of local features: a direct and intuitive perspective lies in the generalization from the concept of Multi-Layered Image (MLI) to the creation of more robust features, extractors and local descriptors. One may implement the MLI into optimization-based systems to mitigate illumination influence and, for example, learning-based techniques as a particular type training paradigm or cost function to help create more robust local feature networks.

— In the mid-level feature planar SLAM module, one could merge the planar-based primitives with other SLAM systems of different geometric primitives such as points or lines. A heterogeneous system seems intuitively much more robust and adaptive in general cases.

— For the relocalization, two future works are imperative: i) it is conductive to explore the intelligent and adaptive control when combining with more layers of

heterogeneous data; ii) more SLAM and robotics applications may be explored by combining the binarized deep feature and the Bag-of-Words technique.

From a long-term vision, three points shall be mentioned here: i) it may entail novel and consistent structures for SLAM systems to be compatible with newly proposed and succeeded deep learning features, instead of the majority of current methods, which often employ deep features as plug-in-like additional constraints; ii) for the illuminative robustness of SLAM systems, a more dynamic and environmental dependent representation of the features or map may be an ambitious yet challenging target. The main idea may consist in synthesizing the modelling, material and lighting condition, also include dynamically estimating and representing the visual information according to different environmental conditions; iii) like the central philosophy of this thesis, treating the SLAM problem per se as an independent problem does gain engineering and research facilities, but it also cuts off the possibility to more organically combine with other robotics, computer vision and even computer graphics tasks. One example is the active SLAMs which consider both motion planning, tracking and mapping: the mapped results can feedback the newer motion planning trajectory for better exploring the space, and motion planning results give more a priori information s.t. the tracking and matching system performs more robustly and precisely.

9.7 Epilogue

SLAM is a complex system composed of multiple heterogeneous types of module: The objective of changing one specific module alone in order to improve the final performance seems insufficient and arduous, as conflicts may interweave and be coupled such that the problem per se is more difficult to be addressed. In a chinese old idiom, this pheonomena is depicted in a narrative called: 'Blind men touching the Elephant'. It tells a story about a king who is chatting with some blind men, they don't know what an elephant is, so he has an elephant brought over and all the blind men touch different parts of it. The one who felt it's tusks said the elephant was the shape of a carrot, the one who felt its ear said that the elephant was like a big but shallow bamboo basket, the one that felt the elephant's leg said it was like a pillar, the one that felt its belly said it was like a urn, and the one that felt its tail said that it was like a rope.

I think this idiom makes an appropriate metaphor for SLAM research: one should have a systematic and allaround understanding rather than seeing through narrow definitions

or trying to overly focus on specific domains. The inspiration of this idiom makes this thesis possible: we try to contribute a bit on a complicated problem, in an overall method by attacking on different sub-domains respectively. We traversed multiple yet critical modules in SLAM system relative to the characteristics of robustness: from local features to geometric primitives, then towards relocalization and image descriptors.

We improved local features on dark condition by proposed multiple layered image structure; we exploited the multiple planar hypothesis as a novel geometric primitive and achieved better tracking and mapping performance; we finally introduced a binary graph descriptor can help SLAM loop detection with heterogeneous type of images layers despite extreme disparities on location appearance.

With these contributions and improvements, we hope to have proposed in the general understanding the problem of SLAM from various angles. Like the blind man who first thought the tail was a rope, but the more he touches, tries, fumbles, even stumbles, like we did in all the research journeys, I truly believe, that he will eventually realize that he is dealing with a thing much bigger, complicated and beautiful, an elephant.

BIBLIOGRAPHY

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, « SLIC superpixels compared to state-of-the-art superpixel methods », *in*: *IEEE Trans. on pattern analysis and machine intelligence* (2012).

[2] Anubhav Agarwal, CV Jawahar, and PJ Narayanan, « A survey of planar homography estimation techniques », *in*: *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12* (2005).

[3] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool, « Night-to-day image translation for retrieval-based localization », *in*: *2019 Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2019.

[4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, « NetVLAD: CNN architecture for weakly supervised place recognition », *in*: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, « NetVLAD: CNN architecture for weakly supervised place recognition », *in*: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2016.

[6] Alberto Argiles, Javier Civera, and Luis Montesano, « Dense multi-planar scene estimation from a sparse set of images », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011.

[7] K. S. Arun, T. S. Huang, and S. D. Blostein, « Least-Squares Fitting of Two 3-D Point Sets », *in*: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (1987).

[8] Sungyong Baik, Hyo Jin Kim, Tianwei Shen, Eddy Ilg, Kyoung Mu Lee, and Christopher Sweeney, « Domain Adaptation of Learned Features for Visual Localization », *in*: *BMVC*, 2020.

[9] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, « Consistency of the EKF-SLAM Algorithm », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[10] Simon Baker and Iain Matthews, « Lucas-kanade 20 years on: A unifying framework », *in*: *Int. J. of computer vision* (2004).

[11] Q. Bateux and E. Marchand, « Histograms-based visual servoing », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2016).

[12] H. Bay, T. Tuytelaars, and L. Van Gool, « Surf: Speeded up robust features », *in*: *European Conf. on Computer Vision (ECCV)* (2006).

[13] Selim Benhimane and Ezio Malis, « Homography-based 2d visual tracking and servoing », *in*: *The Int. J. of Robotics Research* (2007).

[14] P. Bergmann, R. Wang, and D. Cremers, « Online Photometric Calibration of Auto Exposure Video for Realtime Visual Odometry and SLAM », *in*: *IEEE Robotics and Automation Letters (RA-L)* 3 (2 2018).

[15] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart, « The EuRoC micro aerial vehicle datasets », *in*: *The Int. J. of Robotics Research* (2016).

[16] Fernando Caballero, Luis Merino, Joaquin Ferruz, and Anibal Ollero, « Vision-based odometry and SLAM for medium and high altitude flying UAVs », *in*: *Journal of Intelligent and Robotic Systems* (2009).

[17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, « Brief: Binary robust independent elementary features », *in*: *European Conf. on Computer Vision (ECCV)* (2010).

[18] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós, « ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM », *in*: *IEEE Trans. on Robotics* (2021).

[19] G. Caron, A. Dame, and E. Marchand, « Direct model based visual tracking and pose estimation using mutual information », *in*: *Image and Vision Computing* (2014).

[20] Jason Chang, Donglai Wei, and John W Fisher, « A video representation using temporal superpixels », *in*: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2013.

[21] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, « Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation », *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, Cham: Springer Int. Publishing, 2018.

[22] CKCN Chow and Cong Liu, « Approximating discrete probability distributions with dependence trees », *in*: *IEEE Trans. on Information Theory* (1968).

[23] Alejo Concha and Javier Civera, « Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence », *in*: *Proc. of The Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[24] Alejo Concha and Javier Civera, « Using superpixels in monocular SLAM », *in*: *2014 IEEE Int. Conf. on robotics and automation (ICRA)*, 2014.

[25] Mark Cummins and Paul Newman, « Appearance-only SLAM at large scale with FAB-MAP 2.0 », *in*: *The Int. J. of Robotics Research* (2011).

[26] Mark Cummins and Paul Newman, « FAB-MAP: Probabilistic localization and mapping in the space of appearance », *in*: *The Int. J. of Robotics Research* (2008).

[27] Navneet Dalal and Bill Triggs, « Histograms of Oriented Gradients for Human Detection », *in*: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[28] A. Dame and E. Marchand, « Optimal detection and tracking of feature points using mutual information », *in*: *Image Processing (ICIP), IEEE Int. Conf. on*, 2009.

[29] Amaury Dame and Eric Marchand, « Second-order optimization of mutual information for real-time image registration », *in*: *IEEE Trans. on Image Processing* (2012).

[30] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse, « MonoSLAM: Real-time single camera SLAM », *in*: *IEEE Trans. on pattern analysis and machine intelligence* (2007).

[31] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid, « Evaluation of GIST Descriptors for Web-Scale Image Search », *in*: CIVR '09, Santorini, Fira, Greece: Association for Computing Machinery, 2009.

[32] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, « D2-net: A trainable cnn for joint description and detection of local features », *in*: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] J. Engel, V. Koltun, and D. Cremers, « Direct Sparse Odometry », *in*: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2017).

[34] J. Engel, T. Schöps, and D. Cremers, « LSD-SLAM: Large-Scale Direct Monocular SLAM », *in*: 2014.

[35] J. Engel, V. Usenko, and D. Cremers, « A Photometrically Calibrated Benchmark For Monocular Visual Odometry », *in*: 2016.

[36] Pedro F Felzenszwalb and Daniel P Huttenlocher, « Efficient graph-based image segmentation », *in*: *Int. J. of computer vision* (2004).

[37] Martin A Fischler and Robert C Bolles, « Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography », *in*: *Communications of the ACM* (1981).

[38] Martin A. Fischler and Robert C. Bolles, « Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography », *in*: *Commun. ACM* (1981).

[39] Alex Flint, David Murray, and Ian Reid, « Manhattan scene understanding using monocular, stereo, and 3d features », *in*: *Int. Conf. on Computer Vision*, 2011.

[40] G. Florentz and E. Aldea, « SuperFAST: Model-based adaptive corner detection for scalable robotic vision », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[41] David F Fouhey and Adviser Daniel Scharstein, « Multi-model estimation in the presence of outliers », *in*: *Bachelorsthesis, Middlebury College, Middlebury* (2011).

[42] Keinosuke Fukunaga and Larry Hostetler, « The estimation of the gradient of a density function, with applications in pattern recognition », *in*: *IEEE Trans. on information theory* (1975).

[43] Axel Furlan, David Miller, Domenico G. Sorrenti, Li Fei-Fei, and Silvio Savarese, « Free your Camera: 3D Indoor Scene Understanding from Arbitrary Camera Motion », *in*: *BMVC*, 2013.

[44] Dorian Gálvez-López and J. D. Tardós, « Bags of Binary Words for Fast Place Recognition in Image Sequences », *in*: *IEEE Trans. on Robotics* (2012).

[45] Emilio Garcia-Fidalgo and Alberto Ortiz, « iBoW-LCD: An Appearance-Based Loop-Closure Detection Approach Using Incremental Bags of Binary Words », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2018).

[46] S. Gauglitz, T. Höllerer, and M. Turk, « Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking », *in*: *Int. J. of Computer Vision* 94 (2011).

[47] Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieto, and Cesar Cadena, « X-view: Graph-based semantic multi-view localization », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2018).

[48] W Nicholas Greene, Kyel Ok, Peter Lommel, and Nicholas Roy, « Multi-level mapping: Real-time dense monocular SLAM », *in*: *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[49] M. D. Grossberg and S. K. Nayar, « Modeling the space of camera response functions », *in*: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2004).

[50] Banglei Guan, Pascal Vasseur, Cédric Demonceaux, and Friedrich Fraundorfer, « Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions », *in*: *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2018.

[51] Christopher G Harris, Mike Stephens, et al., « A combined corner and edge detector. », *in*: *Alvey vision Conf.* Citeseer, 1988.

[52] Richard Hartley and Andrew Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.

[53] Ming Hsiao, Eric Westman, Guofeng Zhang, and Michael Kaess, « Keyframe-based dense planar SLAM », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[54] Hossam Isack and Yuri Boykov, « Energy-based geometric multi-model fitting », *in*: *Int. J. of computer vision* (2012).

[55] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, « Aggregating local descriptors into a compact image representation », *in*: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2010.

[56] Jianbo Shi and Tomasi, « Good features to track », *in*: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 1994.

[57] Max A. Viergever Josien P.W. Pluim J. B. Antoine Maintz, « Mutual information matching and interpolation artifacts », *in*: *Proc.SPIE* 3661 (1999).

[58] Michael Kaess, « Simultaneous localization and mapping with infinite planes », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[59] James T Kajiya, « The rendering equation », *in*: *Proc. of the 13th annual Conf. on Computer graphics and interactive techniques*, 1986.

[60] Yasushi Kanazawa and Hiroshi Kawakami, « Detection of planar regions with uncalibrated stereo using distribution of feature points », *in*: *In British Machine Vision Conf.* 2004.

[61] Pyojin Kim, Brian Coltin, and H Jin Kim, « Linear RGB-D SLAM for planar environments », *in*: *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[62] Georg Klein and David Murray, « Parallel tracking and mapping for small AR workspaces », *in*: *2007 6th IEEE and ACM Int. symposium on mixed and augmented reality*, 2007.

[63] Nicola Krombach, David Droeschel, and Sven Behnke, « Combining feature-based and direct methods for semi-dense real-time stereo visual odometry », *in*: *Int. Conf. on intelligent autonomous systems*, Springer, 2016.

[64] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl, « A cross-season correspondence dataset for robust semantic segmentation », *in*: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019.

[65] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl, « Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization », *in*: *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, 2019.

[66] Phi-Hung Le and Jana Košecka, « Dense piecewise planar RGB-D SLAM for indoor environments », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[67] Yu Liu, Yvan Petillot, David Lane, and Sen Wang, « Global localization with object-level semantics and topology », *in*: *2019 Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2019.

[68] D. Lowe, « Distinctive Image Features from Scale-Invariant Keypoints », *in*: *Int. J. Computer Vision* (2004).

[69] James MacQueen et al., « Some methods for classification and analysis of multivariate observations », *in*: 1967.

[70] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman, « 1 Year, 1000km: The Oxford RobotCar Dataset », *in*: *The Int. J. of Robotics Research (IJRR)* 36.*1* (2017).

[71] András L Majdik, Charles Till, and Davide Scaramuzza, « The Zurich urban micro aerial vehicle dataset », *in*: *The Int. J. of Robotics Research* (2017).

[72] Ezio Malis and Manuel Vargas, « Deeper understanding of the homography decomposition for vision-based control », PhD thesis, INRIA, 2007.

[73] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler, « Pose estimation for augmented reality: a hands-on survey », *in*: *IEEE Trans. on visualization and computer graphics* (2015).

[74] Christopher Mei, Selim Benhimane, Ezio Malis, and Patrick Rives, « Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors », *in*: *IEEE Trans. on Robotics* (2008).

[75] M. J. Milford, G. F. Wyeth, and D. Prasser, « RatSLAM: a hippocampal model for simultaneous localization and mapping », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[76] M.J. Milford and G. Wyeth, « SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights », *in*: *IEEE Int. Conf. on Robotics and Automation*, 2012.

[77] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, « ORB-SLAM: A Versatile and Accurate Monocular SLAM System », *in*: *IEEE Trans. on Robotics* (2015).

[78] Raúl Mur-Artal and Juan D. Tardós, « ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras », *in*: *IEEE Trans. on Robotics* (2017).

[79] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss, « Robust visual localization across seasons », *in*: *IEEE Trans. on Robotics* (2018).

[80] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, « DTAM: Dense tracking and mapping in real-time », *in*: *Int. Conf. on Computer Vision*, 2011.

[81] Fred E. Nicodemus, « Directional Reflectance and Emissivity of an Opaque Surface », *in*: *Appl. Opt.* (1965).

[82] G. Pascoe, W. Madder, M. Tanner, P. Piniés, and P. Newman, « NID-SLAM: Robust Monocular SLAM Using Normalised Information Distance », *in*: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2017.

[83] R. Paul and P. Newman, « FAB-MAP 3D: Topological mapping with spatial and visual appearance », *in*: *2010 IEEE Int. Conf. on Robotics and Automation*, 2010.

[84] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui, « Towards a simulation driven stereo vision system », *in*: *Proc. of Int. Conf. on Pattern Recognition (ICPR)*, 2012.

[85] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, « Deepwalk: Online learning of social representations », *in*: *Proc. of ACM SIGKDD*, 2014.

[86] Christian Pirchheim and Gerhard Reitmayr, « Homography-based planar mapping and tracking for mobile phones », *in*: *IEEE Int. Symposium on Mixed and Augmented Reality*, 2011.

[87] B. Přibyl, A. Chalmers, and P. Zemčiék, « Feature point detection under extreme lighting conditions », *in*: *Proc. Spring Conf. on Computer Graphics*, ACM, 2013.

[88] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer, « PL-SLAM: Real-time monocular visual SLAM with points and lines », *in*: *2017 IEEE Int. Conf. on robotics and automation (ICRA)*, IEEE, 2017.

[89] Tong Qin, Peiliang Li, and Shaojie Shen, « VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator », *in*: *IEEE Trans. on Robotics* (2018).

[90] Jason Rambach, Paul Lesur, Alain Pagani, and Didier Stricker, « SlamCraft: Dense Planar RGB Monocular SLAM », *in*: *2019 16th Int. Conf. on Machine Vision Applications (MVA)*, 2019.

[91] Urs Ramer, « An iterative procedure for the polygonal approximation of plane curves », *in*: *Computer graphics and image processing* (1972).

[92] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, « The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[93] A. Rana, G. Valenzise, and F. Dufaux, « Evaluation of feature detection in HDR based imaging under changes in illumination conditions », *in*: *IEEE Int. Symposium on Multimedia (ISM)*, 2015.

[94] A. Rana, G. Valenzise, and F. Dufaux, « Learning-Based Tone Mapping Operator for Image Matching », *in*: *IEEE Int. Conf. on Image Processing (ICIP)*, Beijing, China, 2017.

[95] Carolina Raposo and Joao P Barreto, « π Match: Monocular vSLAM and Piecewise Planar Reconstruction Using Fast Plane Correspondences », *in*: *European Conf. on Computer Vision*, Springer, 2016.

[96] Carl Yuheng Ren, Victor Adrian Prisacariu, and Ian D Reid, « gSLICr: SLIC superpixels at over 250Hz », *in*: *ArXiv e-prints* (2015), eprint: `1509.04232`.

[97] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez, « The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes », *in*: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[98] E. Rosten and T. Drummond, « Machine learning for high-speed corner detection », *in*: *European Conf. on Computer Vision (ECCV)* (2006).

[99] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest, « Monocular Vision for Mobile Robot Localization and Autonomous Navigation », *in*: *Int. J. of Computer Vision* 74 (2007).

[100] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, « ORB: An efficient alternative to SIFT or SURF », *in*: *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011.

[101] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys, « Large-Scale Location Recognition and the Geometric Burstiness Problem », *in*: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[102] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al., « Benchmarking 6dof outdoor visual localization in changing conditions », *in*: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[103] Olivier Saurer, Friedrich Fraundorfer, and Marc Pollefeys, « Homography based visual odometry with known vertical direction and weak Manhattan world assumption », *in*: *ViCoMoR 2012: 2nd Workshop on Visual Control of Mobile Robots (ViCoMoR): Workshop: Portugal, in conjunction with the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, 2012.

[104] D. Scaramuzza and R. Siegwart, « Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles », *in*: *IEEE Trans. on Robotics* (2008).

[105] C. Schmid, R. Mohr, and C. Bauckhage, « Evaluation of interest point detectors », *in*: *Int. J. of computer vision* (2000).

[106] P. Seonwook, S. Thomas, and P. Marc, « Illumination Change Robustness in Direct Visual SLAM », *in*: *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[107] C. E. Shannon, « A Mathematical Theory of Communication », *in*: *Bell System Technical Journal* 27.*3* (1948).

[108] Xuesong Shi, Dongjiang Li, Pengpeng Zhao, Qinbin Tian, Yuxin Tian, Qiwei Long, Chunhao Zhu, Jingwei Song, Fei Qiao, Le Song, et al., « Are we ready for service robots? The OpenLORIS-scene datasets for lifelong SLAM », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2020.

[109] G. Silveira, E. Malis, and P. Rives, « An Efficient Direct Approach to Visual SLAM », *in*: *IEEE Trans. on Robotics* (2008).

[110] Sivic and Zisserman, « Video Google: a text retrieval approach to object matching in videos », *in*: *Proc. Ninth IEEE Int. Conf. on Computer Vision*, 2003.

[111] Fabien Spindler, « Vision-based robot control with ViSP », *in*: *ICRA 2018-Tutorial on Vision-based Robot Control*, 2018.

[112] C. V. Stewart, « Bias in robust estimation caused by discontinuities and multiple structures », *in*: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (1997).

[113] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, « A Benchmark for the Evaluation of RGB-D SLAM Systems », *in*: *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.

[114] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford, « On the performance of convnet features for place recognition », *in*: *IEEE/RSJ Int. Conf. on intelligent robots and systems (IROS)*, IEEE, 2015.

[115] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford, « Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free », *in*: *Robotics: Science and Systems XI:* (2015).

[116] C-H Teh and Roland T. Chin, « On the detection of dominant points on digital curves », *in*: *IEEE Trans. on pattern analysis and machine intelligence* (1989).

[117] P. Thevenaz and M. Unser, « Optimization of mutual information for multiresolution image registration », *in*: *IEEE Trans. on Image Processing* (2000).

[118] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl, « Semantic Match Consistency for Long-Term Visual Localization », *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, Cham: Springer Int. Publishing, 2018.

[119] Roberto Toldo and Andrea Fusiello, « Robust Multiple Structures Estimation with J-Linkage », *in*: *Computer Vision – ECCV 2008*, ed. by David Forsyth, Philip Torr, and Andrew Zisserman, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ISBN: 978-3-540-88682-2.

[120] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, « 24/7 place recognition by view synthesis », *in*: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2018).

[121] Philip HS Torr and Andrew Zisserman, « MLESAC: A new robust estimator with application to estimating image geometry », *in*: *Computer vision and image understanding* (2000).

[122] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool, « Seeds: Superpixels extracted via energy-driven sampling », *in*: *European Conf. on computer vision*, Springer, 2012.

[123] E. Vincent and R. Laganiere, « Detecting planar homographies in an image pair », *in*: *Proc. of the 2nd Int. Symposium on Image and Signal Processing and Analysis.* 2001.

[124] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall, « LSD: A fast line segment detector with a false detection control », *in*: *IEEE Trans. on pattern analysis and machine intelligence* (2008).

[125] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang, « Superpixel tracking », *in*: *Int. Conf. on Computer Vision*, 2011.

[126] T. H. Wang and C. T. Chiu, « Low visual difference virtual high dynamic range image synthesizer from a single legacy image », *in*: *IEEE Int. Conf. on Image Processing*, 2011.

[127] Xi Wang, Marc Christie, and Eric Marchand, « Binary Graph Descriptor for RobustRelocalization on Heterogeneous Data », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2022).

[128] Xi Wang, Marc Christie, and Eric Marchand, « Multiple Layers of Contrasted Images for Robust Feature-Based Visual Tracking », *in*: *IEEE Int. Conf. on Image Processing (ICIP)*, 2018.

[129] Xi Wang, Marc Christie, and Eric Marchand, « Optimized Contrast Enhancements to Improve Robustness of Visual Tracking in a SLAM Relocalisation Context », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[130] Xi Wang, Marc Christie, and Eric Marchand, « Relative Pose Estimation and Planar Reconstruction via Superpixel-Driven Multiple Homographies », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[131] Xi Wang, Marc Christie, and Eric Marchand, « TT-SLAM: Dense Monocular SLAM for Planar Environments », *in*: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Xi'an, China, 2021.

[132] D. H. Wolpert and W. G. Macready, « No free lunch theorems for optimization », *in*: *IEEE Trans. on Evolutionary Computation* (1997).

[133] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang, « Robust superpixel tracking », *in*: *IEEE Trans. on Image Processing* (2014).

[134] S. Yang and S. Scherer, « CubeSLAM: Monocular 3-D Object SLAM », *in*: *IEEE Trans. on Robotics* (2019).

[135] S. Yang and S. Scherer, « Monocular Object and Plane SLAM in Structured Environments », *in*: *IEEE Robotics and Automation Letters (RA-L)* (2019).

[136] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer, « Pop-up slam: Semantic monocular plane slam for low-texture environments », *in*: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[137] Guoshen Yu and Jean-Michel Morel, « ASIFT: An algorithm for fully affine invariant comparison », *in*: *Image Processing On Line* (2011).

[138] J. Yuan L. Sun, « Automatic exposure correction of consumer photographs », *in*: *European Conf. on Computer Vision (ECCV)* (2012).

[139] Hui Zhang, Xiangwei Wang, Xiaoguo Du, Ming Liu, and Qijun Chen, « Dynamic Environments Localization via Dimensions Reduction of Deep Learning Features », *in*: *ICSV 2017: Computer Vision Systems*, 2017, ISBN: 978-3-319-68345-4.

[140] Z. Zhang, « A flexible new technique for camera calibration », *in*: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2000).

[141] Z. Zhang, C. Forster, and D. Scaramuzza, « Active exposure control for robust visual odometry in HDR environments », *in*: *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[142] Zhongfei Zhang and Allen R Hanson, « 3D reconstruction based on homography mapping », *in*: *Proc. ARPA96* (1996).

[143] Marco Zuliani, Charles S Kenney, and BS Manjunath, « The multiransac algorithm and its application to detect planar homographies », *in*: *IEEE Int. Conf. on Image Processing 2005*, 2005.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Robustness of Visual SLAM Techniques to Light Changing Conditions

Mot clés : Robotique, Vision par Ordinateur

Résumé : La technique SLAM (Simultaneous Localization And Mapping) se concentre sur la localisation et la récupération de l'environnement et est l'une des fonctionnalités de base de nombreux produits industriels tels que la réalité augmentée, conduite autonome et même le flux de travail cinématographique moderne, eg. le 'prévis'.

De multiples difficultés dans les différentes layers peuvent influencer la performance finale de la tâche SLAM des agents robotiques, car le pipeline est long et compliqué de la physique du monde réel aux informations requises.

Au fur et à mesure que l'appareil photo numérique acquiert les informations du monde physique et les reinterprète au format numérique, i.e. en pixels, de nombreux compromis ont été faits pour s'assurer que l'ensemble du flux de travail est réalisable. De nombreuses solutions sont proposées pour résoudre chaque problème, respectivement, avec les moyens des modèles de probabilité statistiques classiques au moderne deep learning basé sur les données. Cependant, la quête d'amélioration de la robustesse du robot dans des environnements dynamiques et complexes persiste et devient de plus en plus importante et active pour la recherche en robotique d'aujourd'hui. Le besoin est imminent et considéré comme l'un des facteurs les plus impératifs pour déployer des robots de manière omniprésente dans notre vie quotidienne.

Dans ce contexte, cette thèse tente d'aborder une petite goutte dans l'océan du problème de la robustesse du SLAM, mais dans une structure systématique : nous essayons de décomposer le système SLAM en modules différents et inter-influents. Utilisez ensuite le concept de « diviser pour mieux régner » pour répondre aux questions au sein de chaque module et souhaiter contribuer à la communauté et améliorer la robustesse du SLAM.

Avec les objectifs ci-dessus, les contributions de la thèse sont énoncées comme suit pour aborder le problème de robustesse sous plusieurs angles : 1) Du point de vue de l'image, nous avons proposé une structure d'image à plusieurs layers pour améliorer les performances des caractéristiques d'image locales traditionnelles dans des conditions extrêmes. De plus, une méthode d'optimisation sur la recherche linéaire et l'optimisation convexe assistée par information mutuelle sont conçues pour régler les paramètres optimaux avec la structure proposée ; 2) Du point de vue du primitif géométrique, nous avons proposé une estimation de pose relative et un cadre SLAM sous l'hypothèse de plans multiples, respectivement par des méthodes basées sur des caractéristiques de points d'intéret et basées sur des modèles template. Nous avons essayé d'obtenir de meilleures performances de cartographie et de suivi simultanément à l'aide de l'hypothèse planaire plus générale ; 3) Du point de vue de la relocalisation du système SLAM, l'idée est de récupérer les endroits déjà passés par l'agent robot pour éliminer l'erreur d'estimation globale ou lorsque le robot est en état perdu. Nous avons proposé une structure de graphe avec des embedding binaire pour intégrer des informations spatiales et des formats de données hétérogènes tels que des images de profondeur, des informations sémantiques, même des résultats de deep learning etc. La méthode proposée permet aux systèmes robotiques SLAM de se relocaliser avec un taux de réussite plus élevé, même dans des conditions de différentes éclairage et saisonnières.

Title: Robustness of Visual SLAM Techniques to Light Changing Conditions

Keywords: Robotics, Computer Vision

Abstract: The SLAM (Simultaneous Localization And Mapping) technique concentrates on localizing and recovering the environment in a simultaneous way and is one of the core functionalities of many industrial products such as augmented reality, where the device poses should be tracked in real-time; autonomous driving, where one needs to localize the vehicle in a pre-generated map or unknown environment; and even modern filmmaking workflow, where the relative camera position and orientation are critical for post-processing or real-time prevising for directors and actors to visualise the visual effects on the stage.

Multiple difficulties in different levels can influence the final performance of robot agents's SLAM task, as the pipeline is long and complicated from the real world physics to the required information such as agent poses and 3D map, which help us visualize colourful graphics scenes in AR devices or make hard decisions on the highway for autonomous driving.

Many solutions are proposed for addressing each problem, respectively, with the means from classic statistic probability models to the modern data-driven deep neural network. However, the quest of improving the robot's robustness under dynamic and complicated environments perisists and becomes more and more significant and active for nowadays robotics research. The need for improving the robustness of robot agents is imminent and regarded as one of most imperative factors for deploying robots ubiquitously in our daily life.

Under this context, this thesis tries to address a small drop in the ocean of the problem of SLAM robustness, yet in a very systematic view: we try to break down the SLAM system into different and inter-influential modules. Then use the concept of "divide and conquer" for answering possible questions within each module and wishing to contribute to the community and help improve the robustness of SLAM systems under complicated conditions.

With the above objectives, the contributions of the thesis are stated as follows for tackling the robustness problem from multiple angles: 1) from the image feature angle, we proposed a multiple layered image structure for improving the performance of traditional local image features under extreme conditions. Furthermore, an optimization method on linear searching and mutual information assisted convex optimization are designed for tuning the optimal parameters with the proposed structure; 2) from the geometric primitive angle, we proposed a relative pose estimation and SLAM framework under the multiple planar assumption, by keypoint feature-based and template tracker based methods, respectively. We tried to achieve better performance of mapping and tracking simultaneously with the help of a more general planar assumption; 3) from the angle of relocalization of the SLAM system, the idea is to recover the already passed locations of the robot agent for lowering the overall estimation error or when the robot is in lost status. We proposed a binary graph structure for embedding spatial information and heterogeneous data formats such as depth image, semantic information etc. The proposed method enables robotics SLAM systems to relocalize themselves with a higher success rate even under different lighting, weather and seasonal conditions.