3D Semantic SLAM of Indoor Environment with Single Depth Sensor

Vijaya Kumar Ghorpade

Numéro d'ordre : 833

UNIVERSITÉ CLERMONT AUVERGNE

Ecole Doctorale Des Sciences Pour l'Ingénieur

Institut Pascal, UMR 6602 CNRS, Université Clermont Auvergne,
Axe ISPR : Image, Systèmes de Perception pour la Robotique

Thèse présentée pour obtenir le grade de :

DOCTEUR D'UNIVERSITÉ

Spécialité : ÉLECTRONIQUE ET SYSTÈMES

par

Vijaya Kumar Ghorpade

3D Semantic SLAM of Indoor Environment with Single Depth Sensor

Soutenue publiquement le 20 décembre 2017 devant le jury composé de :

Mme Beatriz Marcotegui	PR MINES ParisTech	Rapporteur
M. Olivier Aycard	HDR Université Grenoble Alpes	Rapporteur
M. David Filliat	PR Ensta ParisTech	Examinateur
M. Thierry Chateau	PR Université Clermont Auvergne	Examinateur
M. Paul Checchin	MCF, HDR Université Clermont Auvergne	Directeur de thèse
M. Laurent Trassoudaine	PR Université Clermont Auvergne	Co-Directeur de thèse

Abstract

Intelligent autonomous actions in an ordinary environment by a mobile robot require maps. A map holds the spatial information about the environment and gives the 3D geometry of the surrounding of the robot to not only avoid collision with complex obstacles, but also self-localization and for task planning. However, in the future, service and personal robots will prevail and need arises for the robot to interact with the environment in addition to localize and navigate. This interaction demands the next generation robots to understand, interpret its environment and perform tasks in human-centric form. A simple map of the environment is far from being sufficient for the robots to co-exist and assist humans in the future. Human beings effortlessly make map and interact with environment, and it is trivial task for them. However, for robots these frivolous tasks are complex conundrums. Layering the semantic information on regular geometric maps is the leap that helps an ordinary mobile robot to be a more intelligent autonomous system. A semantic map augments a general map with the information about entities, i.e., objects, functionalities, or events, that are located in the space. The inclusion of semantics in the map enhances the robot's spatial knowledge representation and improves its performance in managing complex tasks and human interaction. Many approaches have been proposed to address the semantic SLAM problem with laser scanners and RGB-D time-of-flight sensors, but it is still in its nascent phase. In this thesis, an endeavour to solve semantic SLAM using one of the time-of-flight sensors which gives only depth information is proposed. Time-of-flight cameras have dramatically changed the field of range imaging, and surpassed the traditional scanners in terms of rapid acquisition of data, simplicity and price. And it is believed that these depth sensors will be ubiquitous in future robotic applications.

In this thesis, an endeavour to solve semantic SLAM using one of the time-of-flight sensors which gives only depth information is proposed. Starting with a brief motivation in the first chapter for semantic stance in normal maps, the state-of-the-art methods are discussed in the second chapter. Before using the camera for data acquisition, the noise characteristics of it has been studied meticulously, and properly calibrated. The novel noise filtering algorithm developed in the process, helps to get clean data for better scan matching and SLAM. The quality of the SLAM process is evaluated using a context-based similarity score metric, which has been specifically designed for the type of acquisition parameters and the data which have been used. Abstracting semantic layer on the reconstructed point cloud from SLAM has been done in two stages. In *large-scale higher-level semantic interpretation*, the prominent surfaces in the indoor environment are extracted and recognized, they include surfaces like walls, door, ceiling, clutter. However, in *indoor single scene object-level semantic interpretation*, a single 2.5D scene from the camera is parsed and the objects, surfaces are recognized. The object recognition is achieved using a novel shape signature based on probability distribution of 3D keypoints that are most stable and repeatable. The classification of prominent surfaces and single scene semantic interpretation is done using supervised machine learning and deep learning systems. To this end, the object dataset and SLAM data are also made publicly available for academic research.

Keywords: Time-of-flight cameras, 3D point cloud processing, noise filters, registration, SLAM, plane detection, segmentation, object recognition, object detection, object classification, machine learning.

Résumé

Pour agir de manière autonome et intelligente dans un environnement, un robot mobile doit disposer de cartes. Une carte contient les informations spatiales sur l'environnement. La géométrie 3D ainsi connue par le robot est utilisée non seulement pour éviter la collision avec des obstacles, mais aussi pour se localiser et pour planifier des déplacements. Les *robots de prochaine génération* ont besoin de davantage de capacités que de simples cartographies et d'une localisation pour coexister avec nous. La quintessence du robot humanoïde de service devra disposer de la capacité de voir comme les humains, de reconnaître, classer, interpréter la scène et exécuter les tâches de manière quasi-anthropomorphique. Par conséquent, augmenter les caractéristiques des cartes du robot à l'aide d'attributs sémiologiques à la façon des humains, afin de préciser les types de pièces, d'objets et leur aménagement spatial, est considéré comme un plus pour la robotique d'industrie et de services à venir. Une carte sémantique enrichit une carte générale avec les informations sur les entités, les fonctionnalités ou les événements qui sont situés dans l'espace. Quelques approches ont été proposées pour résoudre le problème de la cartographie sémantique en exploitant des scanners lasers ou des capteurs de temps de vol RGB-D, mais ce sujet est encore dans sa phase naissante. Dans cette thèse, une tentative de reconstruction sémantisée d'environnement d'intérieur en utilisant une caméra temps de vol qui ne délivre que des informations de profondeur est proposée. Les caméras temps de vol ont modifié le domaine de l'imagerie tridimensionnelle discrète. Elles ont dépassé les scanners traditionnels en termes de rapidité d'acquisition des données, de simplicité fonctionnement et de prix. Ces capteurs de profondeur sont destinés à occuper plus d'importance dans les futures applications robotiques.

Après un bref aperçu des approches les plus récentes pour résoudre le sujet de la cartographie sémantique, en particulier en environnement intérieur. Ensuite, la calibration de la caméra a été étudiée ainsi que la nature de ses bruits. La suppression du bruit dans les données issues du capteur est menée. L'acquisition d'une collection d'images de points 3D en environnement intérieur a été réalisée. La séquence d'images ainsi acquise a alimenté un algorithme de SLAM pour reconstruire l'environnement visité. La performance du système SLAM est évaluée à partir des poses estimées en utilisant une nouvelle métrique qui est basée sur la prise en compte du contexte. L'extraction des surfaces planes est réalisée sur la carte reconstruite à partir des nuages de points en utilisant la transformation de Hough. Une interprétation sémantique de l'environnement reconstruit est réalisée. L'annotation de la scène avec informations sémantiques se déroule sur deux niveaux : l'un effectue la détection de grandes surfaces planes et procède ensuite en les classant en tant que porte, mur ou plafond ; l'autre niveau de sémantisation opère au niveau des objets et traite de la reconnaissance des objets dans une scène donnée. A partir de l'élaboration d'une signature de forme invariante à la pose et en passant par une phase d'apprentissage exploitant cette signature, une interprétation de la scène contenant des objets connus et inconnus, en présence ou non d'occultations, est obtenue. Les jeux de données ont été mis à la disposition du public de la recherche universitaire.

Mots clés : Caméras de temps de vol, nuages de points 3D, filtrage, recalage, SLAM, détection de plans, segmentation, reconnaissance, détection, classification d'objets, apprentissage automatique.

Résumé étendu : contexte et synthèse du mémoire

Contexte

Mon travail doctoral a été réalisé au sein de l'équipe PerSyst (Perception Systems : systèmes de perception). PerSyst est une équipe de recherche de l'Institut Pascal (UMR 6602 CNRS/UBP), unité mixte de recherche du CNRS (Centre National de la Recherche Scientifique), de l'Université Clermont Auvergne et de sigma Clermont. Le CHU de Clermont-Ferrand est également partenaire du laboratoire. PerSyst fait partie de l'axe ISPR (Image, Systèmes de Perception, Robotique) de cet institut, dont les recherches sont centrées sur la perception artificielle, la robotique et la vision par ordinateur. Plus particulièrement, PerSyst développe des approches globales de perception exploitant plusieurs capteurs en vue de la compréhension de scènes, des approches de reconstruction 3D dense d'environnements complexes avec différentes modalités (caméra, LiDAR, radar), des solutions de localisation 2D/3D et de guidage de robots mobiles par approche mono et multisensorielle (caméra, lidar, proprioceptifs), en intégrant éventuellement la cartographie de l'environnement à construire ou des informations de type SIG (Système d'Information Géo-référencé). L'ensemble des activités est historiquement fondé sur une très forte culture de projet mettant en oeuvre des systèmes de perception temps réel sur des plates-formes réalistes et performantes.

Cette thèse a pu être menée grâce à l'aide d'une bourse octroyée par le Laboratoire d'Excellence ImobS3. Plus précisément, ce travail a été soutenu par un programme de recherche du gouvernement français, le Programme d'Investissements d'Avenir, via "l'équipement d'excellence" RobotEx (ANR-10- EQPX-44) et le LabEx IMobS3 (ANR-10-LABX-16-01), mais aussi l'Union Européenne via le programme opérationnel 2007-2013 (European Regional Development Fund - ERDF), et par la région Auvergne.

La localisation et la cartographie simultanées (activité plus connue sous l'acronyme de SLAM pour *Simultaneous Localization And Mapping*) ont été étudiées intensément depuis sa première formulation technique, il y a désormais trois décennies. Les chercheurs spéculent que cette thématique scientifique entre dans une nouvelle phase appelée : "*age of robust-perception*''. Le SLAM a d'abord fait usage de filtres tels que le filtre de Kalman, les filtres à particules, etc. Dans sa phase dite moderne, le SLAM fait usage de nouveaux capteurs de perception pour la cartographie et localisation. Toutefois, les *robots de prochaine génération* ont besoin de davantage de capacités que de simples cartographies et d'une localisation pour coexister avec nous. La quintessence du robot humanoïde de service devra disposer de la capacité de voir comme les humains, de reconnaître, classer, interpréter la scène et exécuter les tâches de manière quasi-anthropomorphique. Par conséquent, augmenter les caractéristiques des cartes du robot à l'aide d'attributs sémiologiques à la façon des humains, afin de préciser les types de pièces, d'objets et leur aménagement spatial, est considéré comme un plus pour la robotique d'industrie et de services à venir (Kostavelis et Gasteratos, 2015). Pour un robot mobile, une carte sémantique (du grec *sēmantikos* pour "sens") est une carte qui contient, en plus des informations spatiales de l'environnement, une classification en classes connues des entités repérées en son sein. Une connaissance approfondie de ces entités, indépendante du contenu de la carte, est disponible par le raisonnement sur une base de connaissances via un moteur de raisonnement associé (Nüchter et Hertzberg, 2008).

Bien que l'importance de la cartographie sémantique ait été admise depuis des décennies, le

travail de recherche mené dans ce domaine en est encore à ses débuts. Cependant, la cartographie sémantique en environment intérieur reçoit plus d'attention qu'en extérieur : les robots de services conçus aujourd'hui sont majoritairement destinés à travailler dans ce contexte. Les approches actuelles pour traiter de ce problème utilisent des capteurs sophistiqués. Dans cette thèse, un système simple, avec un seul capteur de profondeur, a été utilisé dans un environnement d'intérieur qui voit ses conditions d'éclairage relativement peu fluctuer. La caméra, positionnée sur un trépied mobile, capture des images à partir de positions pré-établies dans l'environnement. Ces emplacements pré-établis sont en fait utilisés comme vérité de terrain pour estimer l'écart avec des poses estimées à partir d'un processus SLAM. Ils sont également exploités comme estimation initiale des poses pour un algorithme ICP (Iterative Closest Point). Le processus SLAM fournit un nuage de points 3D de l'environnement reconstruit qui est ensuite interprété sémantiquement à deux niveaux. Au niveau global, des surfaces planes sont extraites et ensuite classées en fonction de leurs propriétés d'orientation. Au niveau local, l'interprétation sémantique est plus spécifique : elle reconnaît les objets de la scène depuis une prise de vue. Les deux niveaux se complètent dans le cadre d'une analyse sémantique de la cartographie pour la navigation intérieure.

Travaux menés

Le travail réalisé au cours de cette thèse s'est déroulé en trois phases. Au cours d'une première phase, la caméra a été calibrée et la nature de ses bruits a été étudiée avec soin. Dans un deuxième temps, l'acquisition de données en environnement intérieur a été réalisé. La séquence d'images ainsi acquises a alimenté un algorithme de SLAM pour reconstruire l'environnement visité. Enfin, au cours de la troisième et dernière phase, une interprétation sémantique de l'environnement reconstruit est effectuée. Les paragraphes suivants fournissent un bref aperçu de chaque chapitre du manuscrit.

Analyse de la littérature : cartographie sémantique à l'aide d'une caméra temps de vol

Un bref aperçu des approches les plus récentes pour résoudre le sujet de la cartographie sémantique est présenté dans le chapitre 2. Selon le type de capteurs utilisés, le type d'environnement considéré et la tâche visée de main, différentes approches SLAM sémantiques sont évoquées. Un accent particulier est mis sur la cartographie sémantique en environnement intérieur et les méthodes associées de l'état de l'art sont présentées en détail. A partir des données fournies par une caméra de temps de vol, cette thèse s'intéresse à la fois à l'analyse d'une seule scène d'intérieur et à la cartographie sémantique à plus grande échelle. La plupart les approches de cartographie sémantique utilisent des scanners laser sophistiqués ou des caméras RGB-D. Elles ont l'avantage par rapport aux caméras à temps de vol de délivrer l'information couleur. Excepté ce point, ces dernières surpassent les premières sur tous les autres aspects : plus grande portée, caractéristiques de bruit bien identifiées, adaptées pour les applications en robotique. Cependant, à notre connaissance, ces capteurs de profondeur n'ont pas été fréquemment employés pour la cartographie sémantique, car la plupart des techniques de reconnaissance d'objets en vision par ordinateur 3D utilisent les descripteurs liés à la couleur.

L'intérêt des capteurs de profondeur est souligné lors de la revue de la littérature. Un nouveau schéma pour la cartographie sémantique est proposé. Comme le capteur de profondeur ne produit que des données de profondeur, dépourvues d'information couleur, un nouveau descripteur de forme a été conçu. Il est uniquement fondé sur la détection de descripteurs.

Notre approche est mise en oeuvre à l'aide d'un capteur de la gamme SwissRanger de la société Mesa Imaging. Cette caméra délivre des données de profondeur sous forme d'images de points 3D et d'images de réflectance associées. L'environnement considéré est un grand couloir, plusieurs pièces de bureaux se répartissant de chaque côté. L'objectif est de reconstruire

l'environnement en utilisant les collections d'images de points, de localiser chaque acquisition dans le modèle reconstruit et d'effectuer une interprétation sémantique. L'étiquetage de la scène avec informations sémantiques se déroule à deux niveaux, l'un dit supérieur et l'autre inférieur (voir Fig. 1). Le premier implique la détection de grandes surfaces planes et procède ensuite en les classant comme des portes, des murs ou des plafonds. Ce travail est présenté au chapitre 8. L'autre niveau de sémantisation opère au niveau des objets et traite de la reconnaissance des objets dans une scène donnée. L'idée est d'imaginer un robot qui navigue dans un environnement, qui identifie d'importants espaces, des pièces, et qui analyse des scènes particulières afin de retrouver certains objets spécifiques pouvant être manipulés plus tard. L'analyse de la scène implique une étape de suppression du bruit dans les données issues du capteur telle que décrite dans le chapitre 3, l'extraction fond/forme pour isoler l'objet, l'évaluation de descripteurs pour chaque groupe d'objets et une étape de reconnaissance de l'objet en fonction de son descripteur en utilisant une technique d'apprentissage (chapitres 6 et 7).

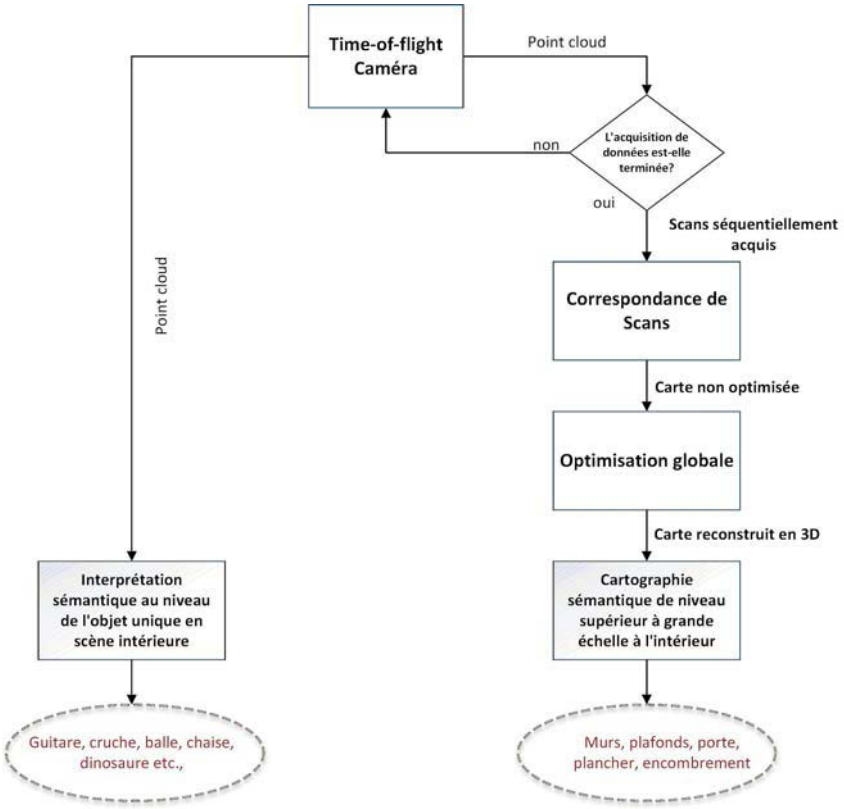


FIGURE 1 – Etapes du processus de cartographie sémantique. L'approche est très semblable à celle de (NÜCHTER et HERTZBERG, 2008), où un étiquetage sémantique des surfaces planes est effectué sur la base de certaines règles puis une détection d'objets utilisant un classifieur préalablement entraîné.

Caméra temps de vol : principe de fonctionnement et filtrage du bruit

Une étape essentielle pour la construction d'une carte 3D de l'environnement est d'ôter le bruit présent au niveau des données brutes qui peut affecter les étapes suivantes du processus de cartographie sémantique. Les données de la caméra du temps de vol souffrent de deux types de bruits différents : des erreurs systématiques et des erreurs non-systématiques. Le chapitre 3 détaille les différents types de bruit présents en sortie du capteur et donne également une brève introduction sur le principe de fonctionnement du capteur. Les *Jump edges* sont des types d'erreurs non-systématiques les plus courantes et les plus importantes et dont l'origine n'est

pas encore complètement comprise. Certains chercheurs suggèrent qu'elles se produisent en raison de réflexions multiples de la lumière incidente tandis que d'autres suggèrent que cela pourrait être dû à la méthode de calcul de la distance par la caméra qui génère des points erronés lorsqu'il y a des surfaces superposées en face de la caméra. Un nouveau traitement fondé sur le principe de fonctionnement du capteur lui-même a été développé pour supprimer les *Jump edges* (voir Fig. 2). Ce chapitre décrit l'approche et évalue les résultats en les comparant à une autre méthode. Elle prend en compte la qualité de l'image filtrée, le temps de calcul et le recalage des images.

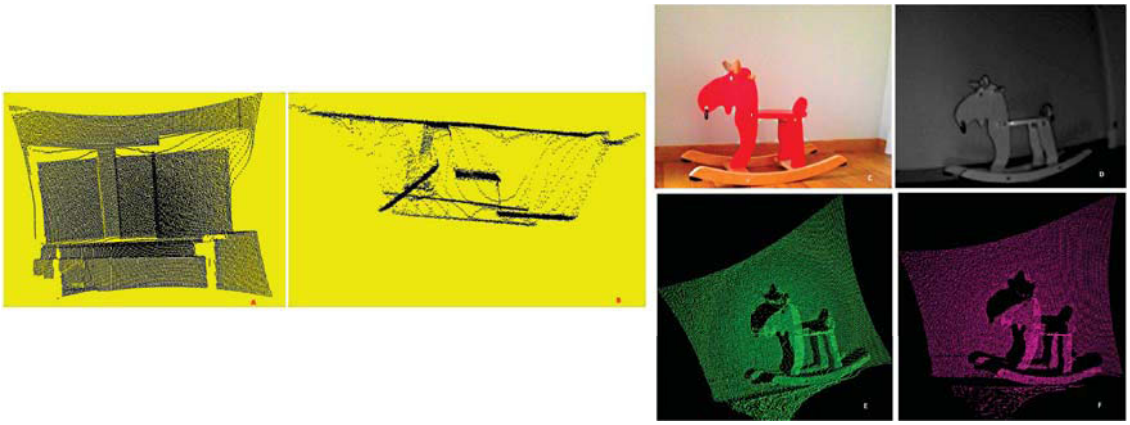


FIGURE 2 – Image de profondeur de trois plans à partir du capteur SwissRanger SR-4000. **A** les *Jump edges* sont observables par les transitions courbes qui vont des surfaces au premier plan vers l'arrière-plan. **B** Vue de dessus montrant les *Jump edges* entre les plans placés à l'avant et ceux à l'arrière. **C** Image RGB test. **D** Image de réflectance. **E** Image de profondeur brute, les *Jump edges* semblent connecter l'objet avec le fond de la scène. **F** Image de profondeur filtrée.

L'algorithme ICP (Iterative Closest Point)

L'une des méthodes les plus populaires pour recaler les relevés de points 3D est présentée dans le chapitre 4. Les algorithmes de recalage expriment les transformations à appliquer à un ensemble de relevés 3D pour les replacer dans un système de coordonnées commun. Le procédé cherche à minimiser l'erreur d'alignement. Partant d'une transformation initiale entre un relevé 3D `Source` et un relevé `Target`, l'algorithme ICP calcule itérativement la transformation jusqu'à ce que l'erreur d'alignement diminue et converge vers une valeur suffisamment faible. Il repose sur l'hypothèse fondamentale selon laquelle les relevés `Source` et `Target` se chevauchent et qu'il existe une correspondance entre ces points. Ce chapitre décrit en détail les variantes de l'état de l'art de l'algorithme ICP. Depuis son invention en 1992, ICP a été exploré intensément. Des centaines de variantes existent aujourd'hui et de nouvelles sont toujours développées. ICP a été mis à profit dans ce travail de thèse pour évaluer les performances des nouveaux filtres de bruit, pour développer une signature de forme invariante à la pose et robuste (voir Fig. 3), pour aligner des relevés 3D pour effectuer un SLAM et obtenir une carte 3D globalement cohérente avec l'algorithme GraphSLAM.

Carte 3D globalement cohérente avec l'algorithme GraphSLAM

Le chapitre 5 commence par un très bref historique des travaux menés en SLAM. Différentes solutions SLAM fondées sur la méthode des moindres carrés sont présentées en allant à l'essentiel. Une discussion plus poussée est menée concernant un système SLAM (exploitant des données 2D et 3D) qui est mis en oeuvre dans cette thèse. Dans les travaux effectués, aucune information issue de centrale inertielle ni d'IMU (Inertial Measurement Unit) ni d'odométrie ni de méthode

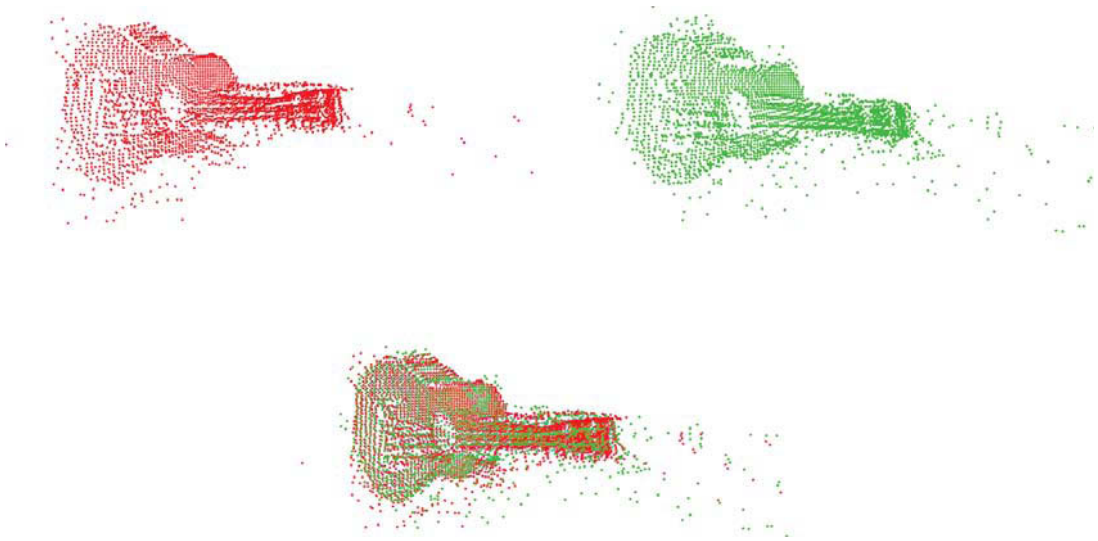
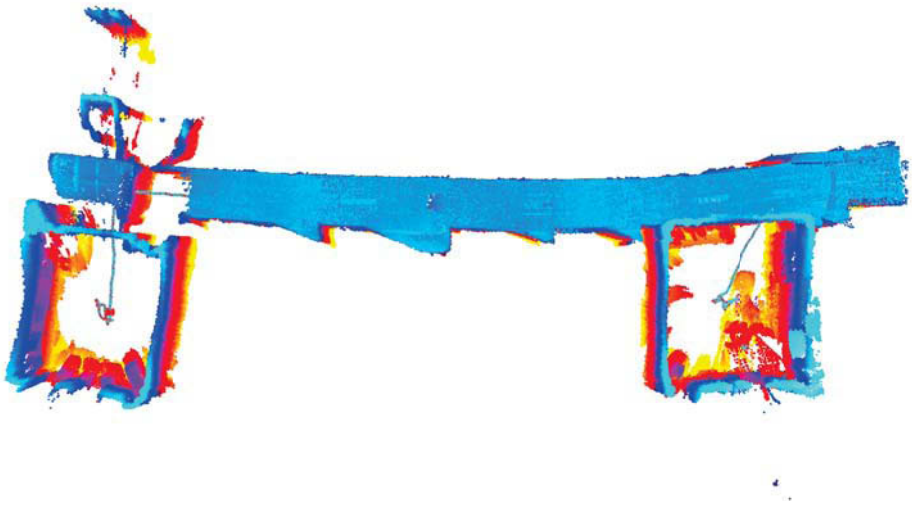


FIGURE 3 – Deux vues différentes d'une guitare, tournées de 5° l'une de l'autre, sont alignées dans le même système de coordonnées en utilisant l'algorithme ICP. A partir d'une vue source et en appliquant la recherche du plus proche voisin dans la vue cible, les points clés similaires sont comptabilisés pour déterminer la répétabilité des descripteurs testés.

testant la fermeture de boucle n'ont été utilisées. Les données sont acquises en plaçant la caméra à des emplacements préétablis de l'environnement intérieur. Ces positions de la caméra sont utilisées comme estimation des poses initiales lors de la mise en correspondance des relevés 3D via la technique d'ICP. Le processus de SLAM produit en sortie un environnement 3D reconstruit (carte sous la forme d'un nuage de points, voir Fig. 4) avec des estimations de pose de la caméra pour chaque acquisition. La performance du système SLAM est évaluée à partir de ces poses estimées en utilisant une nouvelle métrique développée ici. L'extraction des surfaces planes est réalisée sur la carte reconstruite à partir des nuages de points en utilisant la transformation de Hough.

Vers une signature de forme robuste

Comme mentionné précédemment, l'objectif principal de cette thèse est une interprétation sémantique au niveau global de la scène mais aussi au niveau des objets qui la composent. Deux chapitres traitent de ce dernier sujet, l'interprétation sémantique d'une scène depuis une vue. A partir de l'élaboration d'une signature de forme invariante à la pose en passant par une phase d'apprentissage exploitant cette signature, nous sommes parvenus à interpréter une scène contenant des objets connus et inconnus, en présence ou non d'occultations. Ces dernières années, il y a eu une forte croissance dans l'utilisation des modèles 3D en raison du saut technologique permettant de percevoir et visualiser les formes 3D. Cette révolution numérique peut être attribuée à des améliorations substantielles et continues dans les domaines de la microélectronique, de la micro-optique et de la micro-technologie. Les capteurs 3D coûteux qui n'étaient disponibles que dans le cadre d'applications industrielles spécialisées sont désormais disponibles dans des versions largement abordables pour la communauté des chercheurs et le public pour traiter des problèmes de reconstruction 3D, de cartographie, de SLAM, d'interaction homme-machine, de robotique de service. Le jeu, la préservation du patrimoine culturel, la sécurité et la surveillance, l'impression 3D, la CAO sont d'autres utilisations possibles de ces capteurs (SCHÖNING et HEIDEMANN, 2016). Par conséquent, il y a eu une augmentation significative de l'exploitation des représentations 3D pour la détection, la reconnaissance et la classification des objets en s'appuyant sur leur forme. Afin d'identifier un objet quelle que soit sa pose, il faut savoir à quoi ressemble l'objet depuis chaque point de vue de la caméra ou bien avoir le modèle



(a)

(b)

FIGURE 4 – 6DSLAM en utilisant 3DTK. (**a**) 3DTK utilise les positions pré-établies comme estimations initiales des poses et solutionne un SLAM à 6 DoF. Aucune information issue de centrale inertielle ni d'IMU (Inertial Measurement Unit) ni d'odométrie visuelle ni de méthode testant la fermeture de boucle ne sont utilisées. Les données de la caméra SwissRanger sont bruitées et sensibles aux conditions d'éclairage (notamment dans le cas de surfaces vitrées). Les résultats présentés démontrent cet effet. (**b**) Le trajet obtenu par 6DSLAM. Les petites inexactitudes de position par rapport à la vérité de terrain proviennent des bruits de mesure alors que la caméra tourne de 360° entre le début et la fin de la séquence.

3D de l'objet lui-même. Connaître l'objet selon chaque point de vue de la caméra rend le processus de reconnaissance plus facile car la comparaison d'une vue 2.5D avec un modèle 3D est coûteuse en temps et surtout complexe. La reconnaissance en apprenant à quoi ressemble un objet dans chaque pose est beaucoup plus efficace et rapide. L'apprentissage étant un processus hors-ligne unique, la reconnaissance consiste *simplement* à calculer la probabilité d'appartenir à une classe particulière (voir Fig. 5). Pour cela, il est impératif de trouver des relations entre différentes poses du même objet car cela aide le processus de *Machine Learning* à ne pas avoir « *sur-appris* ». Pour ce faire, des points clés, stables vis à vis du changement de pose de l'objet, sont utilisés pour représenter chaque vue. Puis, chaque vue de l'objet est ramenée à une simple distribution de probabilité. La distribution de probabilité capture les relations spatiales et géométriques de ces points clés en utilisant des « fonctions de forme ». Le détecteur de points clés 3D le mieux adapté a été choisi parmi les nombreux existants. Pour effectuer notre sélection, une évaluation de la performance des différents détecteurs a été menée avant la phase de conception de la signature de forme. Deux chapitres (chapitres 6 & 7) présentent cette méthodologie d'évaluation et la conception de la signature de la forme.

Interprétation sémantique au niveau global de la scène

Les connaissances sémantiques de l'environnement à un niveau global, relatives aux espaces fermés, aux murs, aux portes, aux plafonds, aux zones encombrées, sont importantes pour

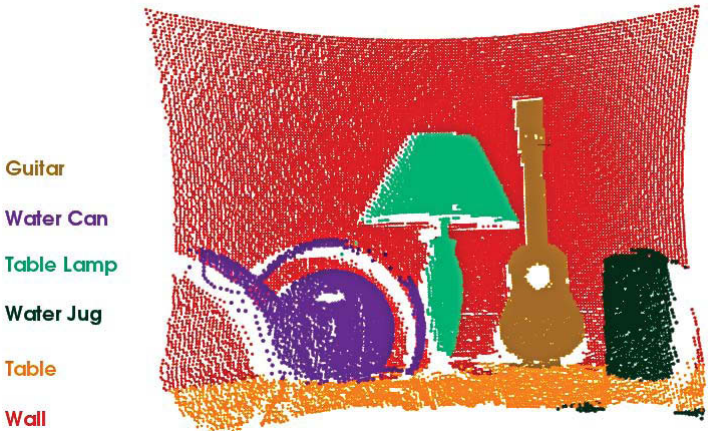


FIGURE 5 – Etiquetage sémantique depuis une vue unique à l'aide de la distribution de points clés. Tous les objets du nuage de points sont correctement reconnus et une couleur associée correspond à leur étiquette. On peut aussi percevoir que même si une partie de la poignée de l'arrosoir n'apparaît pas dans le nuage de points, le modèle est reconnu.

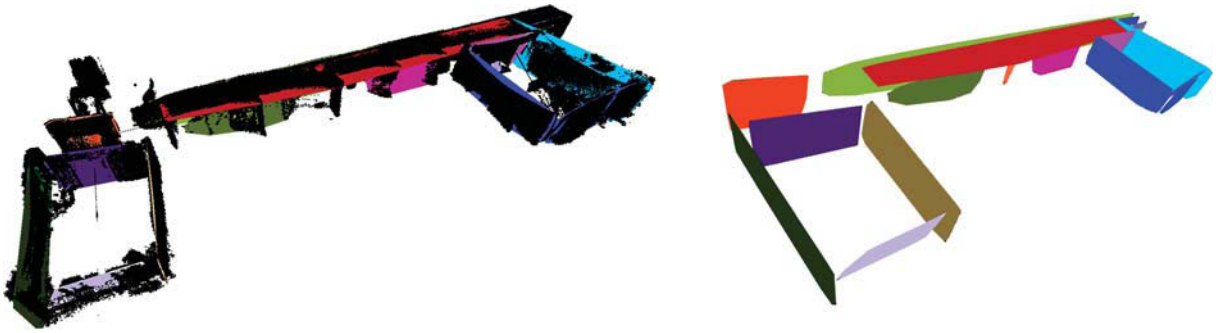


FIGURE 6 – Détection de plans à l'aide de l'algorithme *Randomized Hough Transform*. (**Left**) Nuage de points avec les plans détectés. (**Right**) Plans seuls. Chaque plan est représenté par une couleur distincte (19 plans représentant les murs, les portes, le plafond et le sol). 3DTK fournit également des outils pour extraire des plans des nuages de points. Le nuage de points complet est échantillonné à l'aide d'une méthode à base d'octrees pour un résultat plus rapide et efficace (un point par voxel de 10 cm).

un robot qui doit naviguer et manœuvrer en toute sécurité dans ce contexte (STEINFELD et al., 2006). Une interprétation globale consiste à extraire des surfaces planes, à les classer et à les visualiser. La transformée de Hough est une méthode qui a fait ses preuves pour détecter des objets paramétriques (lignes, cercles, cylindres) et aussi des plans. De nombreuses variantes ont été proposées depuis sa version originelle en 1962. Le chapitre décrit la transformée de Hough *randomisée* (RHT pour Randomized Hough Transform) afin de détecter des plans dans un nuage de points 3D reconstruit (voir Fig. 6). Les plans détectés peuvent être classés en utilisant l'orientation de leur enveloppe convexe dans le référentiel global. Une nouvelle méthode est également présentée dans ce chapitre. Elle exploite les caractéristiques de l'enveloppe convexe, comme l'orientation et la surface des zones planes, qui serviront à classer les surfaces en porte, mur, plafond ou zone encombrée.

Principales contributions

La contribution majeure de cette thèse est d'aborder la problématique de l'analyse sémantique de scènes en environnement d'intérieur dans le cadre d'un processus SLAM réalisé à partir

d'un unique capteur de profondeur. Ci-dessous, quelques détails sur des contributions plus particulières.

- ▷ **Filtrage.** Le capteur de profondeur de type caméra temps de vol souffre principalement d'erreurs non-systématiques (*Jump edges*). Un nouvel algorithme pour éliminer ce type de bruit a été proposé et a été également comparé à une méthode existante.
- ▷ **Evaluation des performances de descripteurs 3D.** Il existe de très nombreux détecteurs de points clés pour les images 2D et 3D. Chaque détecteur possède son propre domaine applicatif spécifique et possède d'excellentes performances pour un type de données et des paramètres particuliers. Afin de concevoir une méthode robuste de reconnaissance d'objets, il faut trouver un détecteur de point clé approprié. Une comparaison de différents détecteurs de points clés 3D est proposée dans cette thèse et le meilleur détecteur identifié a été utilisé pour concevoir le descripteur de forme. Par ailleurs, pour réaliser ce processus d'évaluation, nous avons été amené à développer un jeu de données d'images de profondeur pour différents objets en utilisant un robot cartésien, ce qui a permis d'automatiser la prise d'images autour de l'objet tout en connaissant l'attitude du capteur.
- ▷ **Une nouvelle signature de forme.** Le meilleur détecteur de point clé sélectionné précédemment est utilisé pour représenter un objet depuis une seule vue 2.5D. La relation spatiale entre ces points clés est saisie à l'aide de géodésiques et de distances euclidiennes. Une fonction de distribution de probabilité de ces distances représente de manière unique chaque objet. Un objet peut simplement être reconnu en fonction de ces fonctions de forme. Le système de reconnaissance est développé à l'aide d'un processus d'apprentissage automatique et est exploité pour interpréter une scène capturée depuis un unique point de vue. En outre, un autre jeu de données d'images de profondeur a été spécifiquement conçu pour ce processus.
- ▷ **Jeu de données SLAM.** Les jeux de données publiquement accessibles aident à faire avancer les travaux dans le domaine de la vision par ordinateur, du traitement d'images, de l'apprentissage automatique, de la robotique et bien d'autres domaines scientifiques. Ils permettent l'évaluation scientifique et la comparaison objective des algorithmes avec des indicateurs d'évaluation clairs. Le SLAM est l'un des problèmes de la robotique qui a été étudié en exploitant une variété de capteurs en allant de la caméra 2D ou celle mesurant le temps de vol, en passant par le radar, le sonar et le LiDAR (Cadena et al., 2016). Cependant, à notre connaissance, les imageurs de profondeur n'ont pas été fréquemment exploités pour effectuer un SLAM sémantique. Aussi, on ne trouve pas de jeux de données publics de ce type. Dans cette thèse, nous avons proposé un moyen d'acquérir des données de profondeur en environnement intérieur sans avoir recours à des robots mobiles coûteux. L'ensemble de ces données est mis à la disposition de la communauté. Deux caméras SwissRangers différentes (de portées maximales différentes) ont été mis en oeuvre pour effectuer ce jeu de données constitué d'environ un millier de relevés 3D.
- ▷ **Evaluation des performances du SLAM.** Un logiciel ouvert en accès libre a été utilisé, moyennant quelques légères modifications, pour effectuer un SLAM à partir de nos données. Une évaluation des performances sur les résultats obtenus a été menée selon un nouveau critère qui a été proposé et qui est similaire à l'erreur de pose relative introduite dans (Sturm et al., 2012). Toutefois, nous utilisons des informations contextuelles qui sont intégrées au niveau des scores de similarité. Les poses estimées par le processus SLAM sont comparées à la vérité de terrain (positions préétablies où la caméra est maintenue lors de l'acquisition des images) et un score fondé sur la similarité contextuelle.
- ▷ **Classification des surfaces planes.** La transformée de Hough a été utilisée pour extraire des surfaces planes du nuage de points reconstruit pour un environnement d'intérieur. Ces plans sont délimités par leur enveloppe convexe. Un système à base d'apprentissage classe

ces plans comme étant de type porte, mur, plafond ou autre en utilisant les propriétés d'orientation de l'enveloppe convexe et sa superficie. Le système ainsi conçu a été testé et évalué. Il a fourni, sur les données traitées, de très bons scores.

Mots-clés : Caméra temps de vol, traitement de nuages de points 3D, filtrage, alignement, SLAM, détection de surface plane, segmentation, reconnaissance/détection/classification d'objets, apprentissage machine.

Dedication

For Eliana

Contents

List of Figures

List of Tables

<–

Chapter

1

Introduction

The robots are coming

with clear-cased woofers for heads,
no eyes. They see us as a bat sees
a mosquito a fleshy echo,
a morsel of sound. You've heard
their intergalactic tour busses
purring at our stratosphere's curb,
awaiting the counter intelligence
transmissions from our laptops
and our earpieces, awaiting word
of humanity's critical mass,
our ripening. How many times
have we dreamed it this way- The Age of the Machines,
the post industrial specter
of tempered paws, five welded fingers…

-The Baffler

1.1 Introduction

"The robots are coming", this exclamation or concern was used since the industrial revolution or at least since 1920s. But, the robots are still not around us as we were expecting and depicting in our pop-culture, literature and cinema. May be 1900s and middle of the last century were very optimistic about robots use in everyday life, unaware of the predicaments in this "wicked problem"- you don't understand the problem until you have developed a solution: thus a problem creates more problems. This is the reason why the most popular, commercially available domestic robot is a simple automated vacuum cleaner: *roomba*, while we were expecting fully functional robot maid in next 10 years since 1930 (see Fig. 1.1).

> In the fifties, it was predicted that in 5 years robots would be everywhere.
> In the sixties, it was predicted that in 10 years robots would be everywhere.
> In the seventies, it was predicted that in 20 years robots would be everywhere.
> In the eighties, it was predicted that in 40 years robots would be everywhere.
>
> Marvin Minsky (1927–2016)

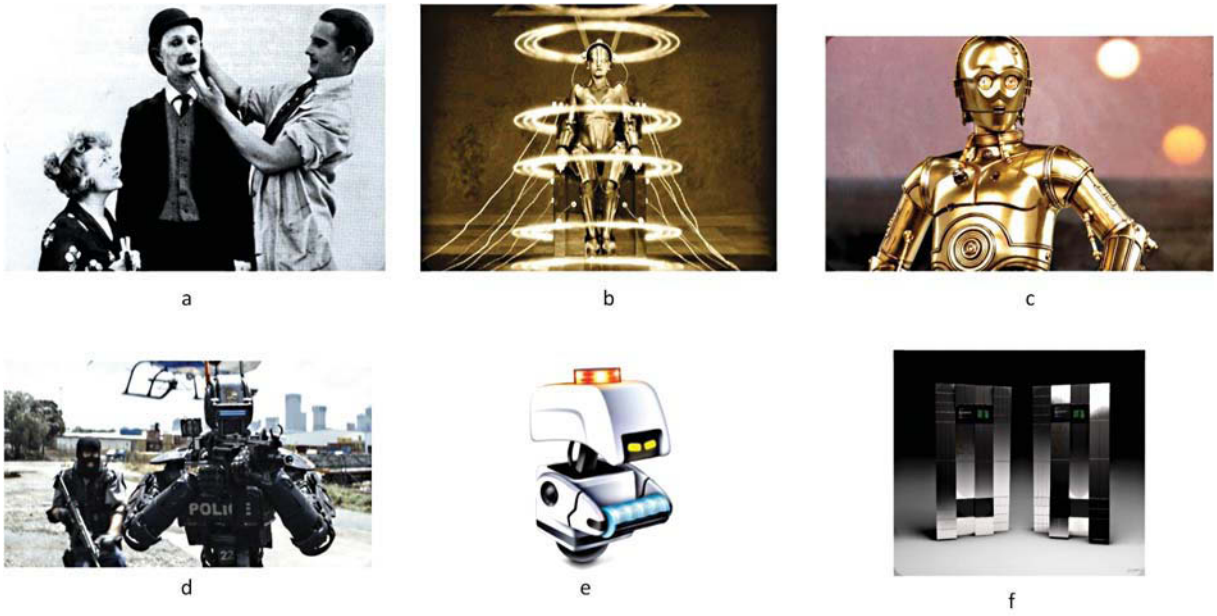


Figure 1.1 – Fictional robots in pop-culture (**a**) A mechanical dummy (before the term "robot" coined) modelled after a janitor in the film ("A Clever Dummy," June 1917). (**b**) The *Maschinenmensch* in the movie ("Metropolis," Jan. 1927) is a gynoid has influenced pop-culture for a long time. The famous droid in StarWars, C-3PO's (c) design is hugely based on it. (**c**) C-3PO from Episode-IV- A New Hope is a humanoid robot intended to assist in customs, etiquettes and translation; has ability to translate over six million langauges/communications ("Star Wars: Episode IV-A New Hope," May 1977). (**d**) Chappie robot in CHAPPiE ("CHAPPiE," Mar. 2015); a droid with human-like feelings (with AI) originally designed for policing the crimes in the city. (**e**) M-O a tiny *cleanerbot* with OCD (Obsessive Compulsive Disorder for cleanliness) in the movie ("WALL-E," June 2008). (**f**) TARS and CASE from the movie ("Interstellar," Nolan, Nov. 2014). TARS is a highly intelligent, witty and humorous Marine Corps tactical robot designed to assist for space travel and data collection and analysis. CASE is a quiet personality Marine surplus robot designed for maintenance and operation of spacecraft Endurance.

These sentences clearly show the unrealistic expectation from computer scientists; unaware of the challenges that will be posed by complex, dynamic and unstructured human environment. However, since past 15 years, there has been explosive growth in domestic and service robots.

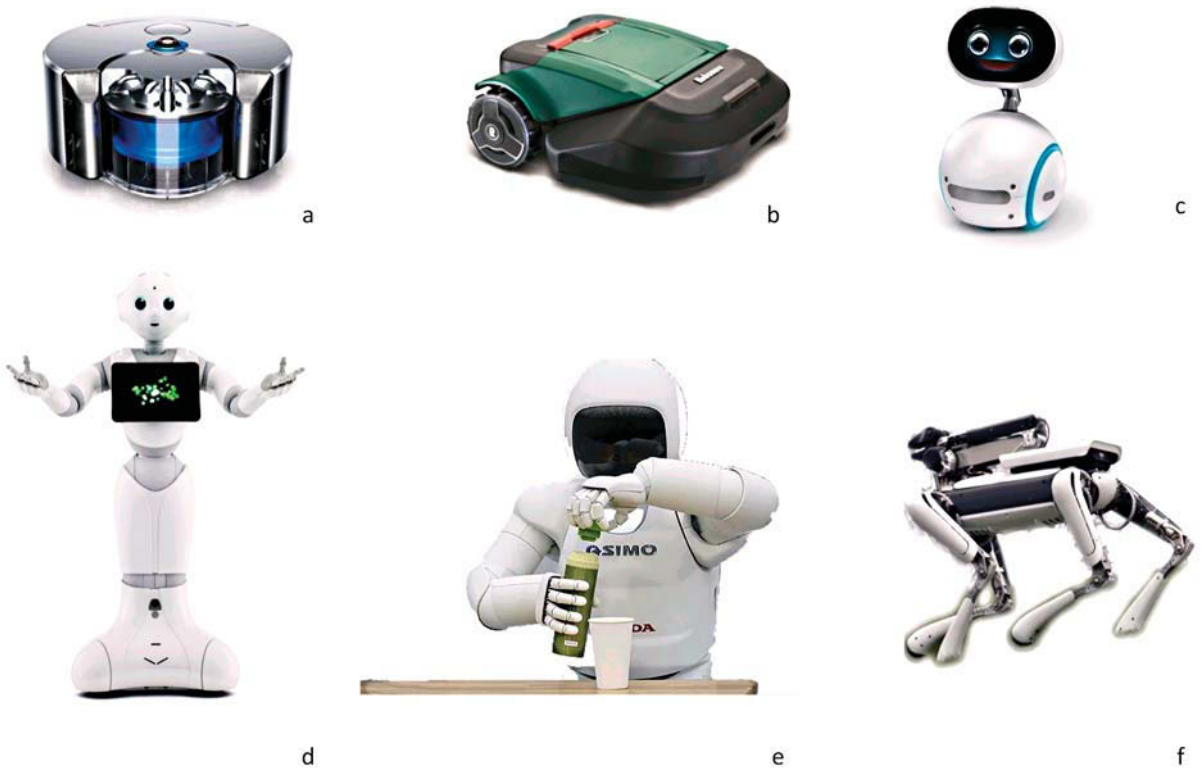


Figure 1.2 – Real robots presently available. (**a**) Dyson's 360 Eye ("Dyson 360Eye"): a home vaccume cleaner like iROOMBA ("Roomba 966"). (**b**) Robomow, a lawn mowing robot ("Robomow"), usually autonomous can be scheduled few time a week to clean up the lawn. However, needs manual guide wires for the task. (**c**) Asus Zenbo ("ASUS Zenbo"), a entertaining little robot which can control home, setup appointments, take picture, speak, listen and even dance. (**d**) Pepper from Softbank robotics ("Softbank Robotics Pepper"), recently became the first humanoid robot to be adopted in Japanese homes. Nao and ROMEO are her friends from same company. (**e**) ASIMO (Advance Step in Innovative MObility) from Honda ("Asimo"). It is presently the most advance robot in the world, with intelligence of a 5-year-old child, can walk up and down stairs, run, carry things, point, wave and push; which are phenomenal physical abilities for humanoid robot. New features are added to it since it's creation from 1986. (**f**) SpotMini from Boston Dynamics ("SpotMini"). Several impressive mobile robots are developed by Boston Dynamics which can walk on difficult terrains, run, jump, carry things. SpotMini is shown in videos to wash dishes in kitchen. Boston Dynamics is recently acquired by SoftBank Robotics.

These robots called as *next generation robots* (see Fig. 1.2) should not only have to track their location and navigate between points in space, but also reason, interpret and acquire knowledge about space, plan tasks and interact with people naturally. The robots presently deployed in real-world human environments have limited capabilities (Fig. 1.2) and far from the desired fictional ones (Fig. 1.1). The most advanced robots presently being Asimo ("Asimo") (see Fig. 1.2 (e)), PR2 ("PR2"), Home Assistant (Yamazaki et al., 2009), PETMAN ("SpotMini") and Nao ("Softbank Robotics Pepper"). There are many initiatives from governments ("Robotic Visions to 2020 and Beyond: The Strategic Research Agenda for Robotics in Europe. European Robotics Technology Platform (EUROP), 2009"; *National Science Foundation : Where Discoveries Begin*; "DoD Announces Award of New Advanced Robotics Manufacturing (ARM) Innov"; Dunbar, 2015; Gelin and Christensen, 2014; Plöger and Nebel, 2008; "Research & Innovation") and private sectors to make robots ubiquitous in next few years if not decades. Visionary, philanthropist and founder of Microsoft Mr. Bill Gates said that next hot field will be robotics after PC's revolution

Figure 1.3 – A Robot in every home by Bill Gates. Source: (Gates, 2007).

(see Fig. 1.3).

Recent advances in computer vision, artificial intelligence and cognitive robotics can be attributed to the objective:

> *To construct physically instantiated systems that can perceive, understand and interact with their environment, and evolve in order to achieve human-like performance in activities requiring context (situation and task) specific knowledge.*
>
> CoSy (*Cognitive Systems for Cognitive Assistants*)

While Beetz et al. (2007) introduced *Assistive Kitchen*, a comprehensive demonstration and challenge scenario for technical cognitive systems, where in selected research subjects are analysed to identify the needed cognitive abilities. Whereas Metta et al. (2010) developed an open-system which promote collaborative research in inactive artificial cognitive system: *iCub*. *iCub* is

a humanoid robot, 94 cm, with 53 degrees-of-freedom, is able to crawl on all fours and sit up, its hands will allow dexterous manipulation, and its head and eyes are fully articulated. It has visual, vestibular, auditory and haptic sensory capabilities. With the complete system being free, the research communities can easily replicate, customize and improve. Several attempts have been made to design integrated cognitive architectures and implement them on mobile robots (Yamazaki et al., 2009; Landsiedel et al., 2017; Spexard et al., 2006; Hawes et al., 2011; *Cognitive Systems for Cognitive Assistants*; Roncone et al., 2016; Martinez-Hernandez et al., 2016; Breazeal, Dautenhahn, and Kanda, 2016; Beetz et al., 2007; Neumann et al., 2017; Schiffer, 2016). These attempts are focussed on creating much better, more versatile systems than the present commercially available robots. These systems need to interact with the environment, interpret the space, move with agility. Spatial understanding is a must for the future robots to perform basic tasks such as navigation, obstacle avoidance, grasping/manipulation and long term autonomous exploration. Spatial knowledge is fundamental for basic human knowledge (Kuipers, 2000). Spatial metaphors are ubiquitous in discourse, and draw on pre-existing spatial knowledge to communicate relationships and processes that would be difficult to communicate otherwise. Spatial knowledge is grounded in sensorimotor experience. It exists in a number of different forms, including procedures for getting from one place to another, topological network maps of an environment, and geometrical models of the environment (Harnad, 1990; Lynch, 1960; Lakoff and Johnson, 2008).

Different types of spatial knowledge can be identified depending on the source, point of reference, spatial scale or level of abstraction (Pronobis, 2011). Geometric aspects of space can be represented by a metric map (Dissanayake et al., 2001; Wolf, Burgard, and Burkhardt, 2005; Paz et al., 2007; Milford and Wyeth, 2008), a level of abstraction on metric space into discrete units lets us focus on spatial topology (topology map) (Ulrich and Nourbakhsh, 2000; Siagian and Itti, 2007; Montemerlo and Thrun, 2007; Cummins and Newman, 2008). Hybrid of metric and topological maps are gaining popularity allowing for better scalability, easier access and maintenance in large-scale environments (Christensen, Kruijff, and Wyatt, 2010). The inclusion of semantics in the map enhances the robot's spatial knowledge representation and improves its performance in complex tasks and human interaction. For example, the task for a robot maid to bring a can of soda from refrigerator becomes easier if it knows the refrigerator is usually in the "kitchen". The assignment of attribute "kitchen" to a particular space in environment rather than a simple room (like in metric or topological map) enhances the spatial knowledge in a more meaningful way (Wu, Lenz, and Saxena, 2014). The semantic map can extend the robots capabilities in performing fundamental and traditional tasks such as navigation, localization, exploration and manipulation (Galindo et al., 2005; Rottmann et al., 2005; Stachniss, Mozos, and Burgard, 2006; Dang and Allen, 2014).

1.2 Towards 3D Semantic SLAM

Although enriching existing maps with semantic information has several uses and mandatory for future robots, it has not yet received the due attention it deserves due to its complexity and challenging real-time solutions. In this thesis, an endeavour to solve some of theexisting problems pertaining to semantic mapping[1] is made. Solving semantic SLAM with a noisy sensor which outputs colorless depth data has never been addressed to the best of our knowledge. Although the sensor generates ramification of problems, it has several advantages which other sensors lack. With the uncertainty in SLAM being fully solved in every environment with any possible sensors, this thesis is towards achieving "3D semantic SLAM for indoor navigation". An office environment is considered for building the map, and few objects are strategically placed to

[1]In this dissertation, "semantic SLAM" and "semantic mapping" terms are used as synonyms. Although the former has no proper definition and might involve simultaneous 3D reconstruction, material recognition and segmentation. And semantic mapping typically involves first 3D reconstruction and then semantic layering upon it.

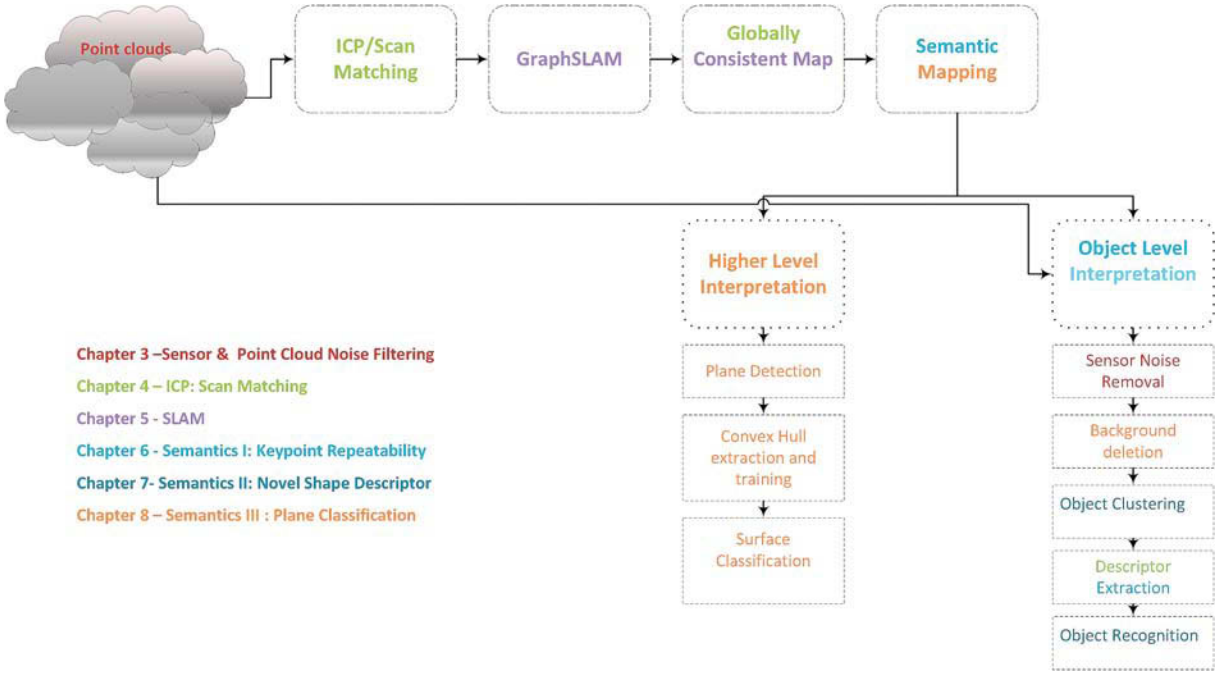


Figure 1.4 – The stages of "Semantic Mapping of Indoor Environment". The font color of each stage matches with chapter number mentioned in the table. Mixed color is used to showcase the use of particular stage (process) in multiple chapters. The point cloud from the sensor is very noisy which is processed in Chapter 3 (red) before applying ICP or object recognition step.

be interpreted. The semantic interpretation is achieved at two levels: higher level, involves the larger objects like walls, ceilings, doors, etc.; and object level, wherein, a particular scene having several objects are interpreted by the system (see Fig. 1.5). The goal being, to ask a robot to make complete reconstruction of the environment and recognize a particular space (like kitchen or office or study room) and then interpret every single 2.5D view from the camera and perform the assigned task. To this extent, several household objects are trained to be recognized with a multi-class classification system which is based on novel shape signature.

1.3 Thesis Outline

In this PhD dissertation, each chapter tries as much as possible to be self-sufficient, without the need to cross-refer. An earnest effort has also been made to give proper external references for the reader to explore related research articles. Permissions from the authors of the original articles has been explicitly taken and mentioned where needed.

1.3.1 Literature Review: Semantic Mapping with Single Depth Camera

A brief survey of the most recent approaches to solve semantic mapping is presented in Chapter 2. It also discusses classification of the semantic SLAM approaches based on sensors utilized, environment considered and task at-hand. A special emphasis on indoor semantic mapping is made in this chapter, and associated state-of-the-art methods are presented in detail. According to this classification system, this thesis addresses indoor single scene and large scale semantic mapping using time-of-flight camera. Most of the semantic mapping approaches utilize sophisticated laser scanners or RGB-D cameras. RGB-D cameras have the upper edge over time-of-flight depth cameras only in providing color information, except this, the latter dominates in every other aspects: has better, longer range and well-studied noise characteristics, specifically designed for robotic applications. However, to the best of our knowledge, these depth sensors have not been

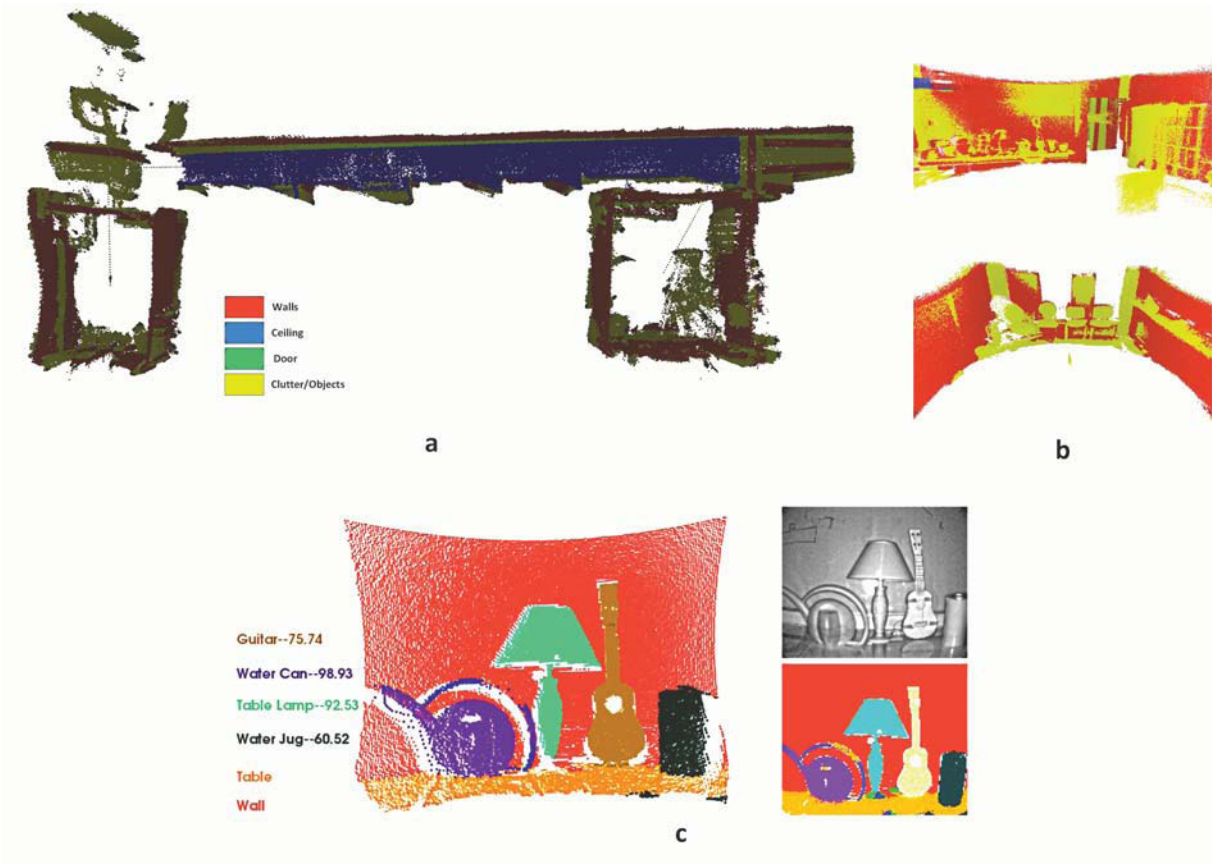


Figure 1.5 – Semantic Mapping of Indoor Environment. (**a**) Higher level interpretation. Planes are classified using a machine learning system based on the convex hull orientation. It perfectly classified walls, ceiling, doors and clutter/objects. (**b**) Objects (chairs) and clutter in Room1 (bottom) extracted, (top) objects in Room2 which are part of machine learning system to be interpreted. (**c**) Semantic interpretation of 2.5D scene of (b, top), using a novel shape descriptor.

considered enough for semantic mapping, as most of the object recognition techniques in 3D Computer Vision use color-based descriptors.

The importance of depth sensors has been realized in this thesis through this literature review, and a new semantic mapping pipeline is proposed. As the depth sensor output only colorless depth data, a new shape descriptor has been designed which is only based on the keypoint detectors that does not need color information.

Our approach uses a SwissRangers' Time-of-Flight sensor, which produces only depth data in the form of point cloud and amplitude images. The environment considered is a large indoor office arena; having several rooms on either side of long corridor. The goal is to reconstruct the environment using the individual sequentially acquired point cloud scans, localise every acquisition in the global reconstructed model and perform semantic interpretation (see Fig. 1.4). Labelling with semantic information is done at two stages: higher level and lower level (see Fig. 1.5). Former involves detecting large planar surfaces and classifying them into doors, walls, ceilings (presented in Chapter 8). The latter or object-level deals with recognition of objects in a given scene. It is like, a robot that is navigating through an environment, it recognized important and large spaces and is analysing individual scenes for some specific objects to be manipulated later on. The object level scene parsing involves: sensor noise removal as dealt in Chapter 3, background/clutter deletion, object extraction, descriptor evaluation for each object cluster and recognizing the object's cluster based on its descriptor using machine learning (Chapter 6 and 7).

1.3.2 Time-Of-Flight Camera: Working Principle and Noise Filtering

The integral and imperative step for constructing a good 3D map of the environment is to free the data from noise which can, otherwise, affect every step in the semantic mapping pipeline. The data from the time-of-flight camera suffer from two different types of noises: systematic and non-systematic errors. This chapter details different types of noise present in the sensor and also gives brief introduction about the working principle of it. *Jump edges* are the most common and prominent type of non-systematic errors whose origin is still not yet fully understood. Some researchers suggest that they occur due to multiple reflections of the incident light while some others suggest that it could be due to averaging algorithm present in the camera software, which generates false points when there are overlapping surfaces in front of the camera. A new algorithm based on the working principle of the sensor itself has been developed to remove jump edges. This chapter describes the approach and evaluates the results with another method in terms of quality of output image, computation time and registration of scans.

1.3.3 ICP: Iterative Closest Point Algorithm

One of the most popular and *de facto* method to register scans is presented in Chapter 4. Registration algorithms associate a set of scans into common coordinate system by minimizing the alignment error. An initial transformation is created to represent the `Source` scan in the same coordinate as the `Target` frame. In ICP, the transformation is iteratively calculated, until the alignment error decreases and eventually converges. It works on a fundamental assumption that the source and target scans are identical, and there exists correspondence between every point in them. This chapter describes state-of-the-art variants of ICP and its counterparts, in detail. Since its invention in 1992, ICP has been explored thoroughly and hundreds of its variants exist today and still new variants are always developed. ICP has been used profusely in this thesis for evaluating novel noise filter performance, developing a robust and pose invariant shape signature, scan registration for SLAM and globally consistent map with GraphSLAM.

1.3.4 SLAM

This chapter starts with a very brief history and origin of SLAM (Simultaneous Localization And Mapping) problem. Different types of SLAM systems based on Gauss' Least Squares method are presented in gist and a very elaborate discussion is given about a SLAM system (for 2D and 3D) on which this thesis is based on. In this thesis, no IMUs (Inertial Measurement Unit) or visual odometry techniques or loop closing methods have been used. Data are acquired by placing the camera at pre-arranged locations on virtual coordinate system created in the indoor environment. These "*locations*" are used as initial pose estimate for ICP scan matching. The SLAM system outputs a reconstructed 3D environment (point cloud) with pose estimates of camera at each acquisition. The SLAM system's performance is evaluated on these pose estimates using a novel metric developed on course. Plane extraction is achieved on the output point cloud using Hough transform.

1.3.5 Towards a Robust Shape Signature: Semantics Part I and Part II

As mentioned earlier, in this thesis higher-level and object-level semantic interpretation is the main objective. These two chapters deal with the latter: single scene semantic interpretation; starting from developing a pose invariant shape signature to learning of this signature and finally interpreting scene having known, unknown objects with and without occlusion. There has been an explosive growth in the usage of 3D models in recent years due to quantum jump in 3D sensing technology to model, digitize and visualize 3D shapes. This digital revolution can be attributed to substantial and continuous improvements in microelectronics, micro-optics and micro technology. These expensive 3D sensors which were once only available for specialized

industrial applications are now commercially available for research communities and public for 3D reconstruction, mapping, SLAM, human-machine interaction, service robotics, gaming, preserving cultural heritage, security and surveillance, 3D printing, CAD and others (Schöning and Heidemann, 2016). As a direct result of this, there has been an exponential increase in the amount of 3D models usage and wherefore determining the similarity between 3D models has become crucial and is also at the core of shape-based object detection, recognition and classification. In order to interpret an object in any pose, it is necessary to know how the object looks from every camera viewpoint or to have the 3D model of the object itself. Knowing the object from every camera viewpoints makes the recognition process easier, as comparing a 2.5D view with 3D model is time taking and complex. Recognition from learning how an object looks in every pose is much more effective and fast, as learning is a single step off-line process, and recognizing is just calculating the *certainty* of belonging to particular class from the machine learning (ML) model. It is imperative to find relations between different poses of the same object, as it helps the ML model not to be too "overfitted" and also it is basis for human cognition. In order to achieve this, keypoints, which are stable across pose changes on object surface, are used to represent each view and each object view is abstracted to simple probability distribution. The probability distribution captures the spatial and geometric relations of these keypoints using some "shape functions". It is also implied to use the best 3D keypoint detector for this purpose, as there are dozens of them. To this end, a performance evaluation of different 3D keypoint detectors is done prior to the design of shape signature. These two chapters present this evaluation methodology and design of the shape signature.

1.3.6 Semantics: Part III

Higher level semantic knowledge of the environment like enclosed spaces, walls, doors, ceilings, clutter are equally important for the robot to navigate and manoeuvre safely around humans (Steinfeld et al., 2006). Higher level interpretation involves extracting planes, classifying and visualizing them. The Hough Transform is an established method for detecting parametrized objects (lines, circles, cylinders) and also for planes. Many variants of it evolved since its invention in 1962. This chapter describes the application of Randomized Hough Transform (RHT) for plane detection in 3D reconstructed point cloud. The detected planes can be classified using their convex hulls orientation in the global reference frame. A new method is also presented in this chapter which learns the convex hull features like orientation and area of planes to classify the surfaces into doors, walls, ceiling and clutter.

1.3.7 Conclusion

The last chapter presents a brief overview of the thesis' set goals and its achievements; and also presents the future work that could be done to extend this work. The main contribution in this thesis is to achieve semantic SLAM with minimalistic equipments, hence only single depth sensor has been used. The other approaches utilize sophisticated devices for mobility, data acquisition and for augmenting the data with additional sensors. We make an endeavour in this thesis to show that, it is possible to do semantic mapping with single depth sensor. This chapter weighs the set goals against the realizations on an empirical balance, and proposes possible extensions.

1.4 Main Contributions

The global major contribution of this thesis is to address the indoor semantic SLAM with a single depth sensor. The following are the sub-contributions.

- ▷ **Noise Filtering** The time-of-flight depth sensor suffers from one of the prominent non-systematic noise, the jump edges. A novel algorithm to remove this noise has been proposed and also compared with the other state-of-the-art method.

- ▷ **Performance Evaluation of 3D Keypoints** There are more than two dozen of keypoint detectors already exists for 2D and 3D data. Each detector has its own specific application and has the best performance for the given data and given set of parameters. In order to design a robust object recognition method, a suitable keypoint detector should be found; the object recognition method actually involves a novel shape descriptor that uses keypoint detector. A new way to compare different 3D keypoints detectors is proposed in this thesis and the resultant best detector is used to design the shape descriptor. Another contribution in the same evaluation process is the development of object depth dataset using a Cartesian robot.
- ▷ **Novel Shape Signature** The best keypoint detector which is found from the above evaluation is used to extract keypoints and represent a single 2.5D view of the object. The spatial relation between these keypoints are captured using geodesics and Euclidean distances. A PDF (Probability Distribution Function) of these distances uniquely represent every object. An object can simply be recognized based on its keypoint's spatial PDFs. The recognition system is developed using machine learning and applied to interpret single scene. Also, another object dataset is specifically made for this process.
- ▷ **SLAM Dataset** Publicly available and benchmark datasets help to push forward the state-of-the-art techniques in Computer Vision, Image Processing, Machine Learning, Robotics and several other scientific domains. They support the scientific evaluation and objective comparison of algorithms with a clear evaluation metrics. SLAM is one of the problems in robotics which has been investigated using a variety of image and time-of-flight sensors that use radar, sonar and LiDAR (Cadena et al., 2016). However, to the best of our knowledge, depth sensors have not been exploited enough for semantic SLAM, as a result we lack standard publicly available dataset for it. In this thesis we proposed a way to acquire depth data of indoor environment without the need for expensive mobile robots and also the dataset is made publicly available for academic research. The dataset has utilized two different SwissRanger cameras with different maximum ranges and has around one thousand scans.
- ▷ **Evaluation of SLAM** A publicly available software (with slight modifications) has been used to solve the problem of SLAM. The performance of the software for our dataset is evaluated using a novel way similar to relative pose error (Sturm et al., 2012), however, we use context information embedded in the cosine similarity scores. The SLAM's pose estimates are evaluated against the ground truth (the prearranged positions where the camera is kept while data acquisition) using this novel context-based similarity score.
- ▷ **Surface Classification** Hough Transform has been used for extraction of planes from the reconstructed point cloud of indoor environment. The output of Hough Transform are the planes represented using convex hulls. A novel machine learning system has been decided to classify the planes into doors, walls, ceiling and clutter using the orientation properties of convex hull and its area. The designed system is almost 100% accurate.

1.5 Funding

This thesis has been carried out with the help of a grant awarded by the Laboratory of Excellence ImobS3. More precisely, this work was supported by a French government research program, the Investments for the Future Program, via "l'Équipement d'Excellence" RobotEx (ANR-10-EQPX-44) and LabEx IMobS3 (ANR-10-LABX-16-01), and also with the European Union via the program 2007-2013 (European Regional Development Fund - ERDF), and by the Auvergne region.

Chapter

2

Literature Review: Semantic Mapping with Single Depth Camera

After emphasising the semantic stance in regular maps and its indispensability for future service robots in previous chapter, this chapter presents the state-of-the-art methods that have been developed and applied to solve this problem. The need for "semantics in robots" although has been recognized almost four decades ago, it's still in its nascent phase. Starting with a proper definition of this problem and its relevance in SLAM (Simultaneous Localization And Mapping), different categories of semantic mapping based on types of sensors used in a particular environment (indoor or outdoor) is discussed in this chapter. Semantic interpretation can be done at two scales: single 2.5D scene or on a large-scale 3D reconstructed environment using either laser scanners or RGB-D sensors. However, depth sensors have not been explored enough to approach this problem. In this thesis, both these scales are explored using single depth camera: interpretation of single scene involving object level detection/recognition and higher-level interpretation of large-scale environment. The latter involves labelling of large surfaces and key components in the indoor facility.

2.1 Introduction

One of the most important attribute for a future service robot to co-exist with humans, is to have cognition and understanding of its environment. The quintessential humanoid/service robot should have impeccable ability to see like humans see, able to recognize, classify, interpret the scene and perform the tasks in human-centric form. Hence, augmenting the robot's maps architecture with semiological attributes involving human concepts, such as types of rooms, objects and their spatial arrangement, is considered as a must do for future service robotic industry (Kostavelis and Gasteratos, 2015). The enhancement of regular maps with semantic information about the environment is called *semantic*[1] mapping.

> *A semantic map for a mobile robot is a map that contains, in addition to spatial information about the environment, assignments of mapped pertaining to entities of known classes. Further knowledge about these entities, independent of the map contents, is available for reasoning in some knowledge base with an associated reasoning engine. Source: (Nüchter and Hertzberg, 2008).*

Vision is the most dominant modality for robot navigation, localization and mapping (Sibley et al., 2010; Agrawal and Konolige, 2008; Milford, Wyeth, and Prasser, 2004; Ulrich and Nourbakhsh, 2000; Cummins and Newman, 2008; Davison et al., 2007; Harris and Pike, 1987; Neira et al., 1997; Sim, Elinas, and Griffin, 2005; Maddern, Milford, and Wyeth, 2012; Murillo et al., 2013; Latif et al., 2014; Neubert, Sünderhauf, and Protzel, 2015). The first twenty years of vision-based robot navigation, surveyed in (Desouza and Kak, 2002), state that any *function-driven navigation*, such as to locate an object and bring it, is to be associated with the overall problem of computer vision, *i.e.*, automatic scene interpretation. However, in order for a robot to be able to navigate efficiently, a consistent geometrical map should be the first foundation in the architecture. It can be analogised with construction of a house, where the foundation, the basement of the house is the core which should be laid first and then one can think of fixing a mural. Several decades of laborious research has been conducted in SLAM (Simultaneous Localization and Mapping), considered to be "*chicken and egg*" problem (Thrun, Burgard, and Fox, 2000), which led to fruitful and remarkable results (Thrun, Burgard, and Fox, 2005). The representative works described in (Thrun, Burgard, and Fox, 2005; Jian et al., 2013; Grisetti, Stachniss, and Burgard, 2007; Hähnel et al., 2003) prove the necessity for an accurate representation of the robot's surroundings as well as the development of efficient mapping methods. With SLAM being solved, researchers can focus more on fixing walls, windows, mural and interior design. No progress could have been made in the area of semantic mapping, unless a prior advancement in SLAM had been made.

A deeper understanding of SLAM requires a further decomposition of the problem. A taxonomical classification of mapping methods results in three classes, viz. the metric, topological and topometric. Metric mapping (please refer to Fig. 5.5 for different types of 3D metric maps) is a geometrical representation, where all the poses are relative to the global coordinate system. Typically, it is either 3D map or 2D occupancy grid. An occupancy grid map represents the environment as a block of cells, each one either occupied, so that the robot cannot pass through it, or unoccupied, so that the robot can traverse it. Topological maps on the other hand involves a graph, each of the nodes corresponds to particular location in the real world (Thrun, 1998) (Angeli et al., 2009). Topometric (Cowley, Taylor, and Southall, 2011), as the name says, is a combination of both metric and topological maps, it facilitates faster and more accurate robot localization, (Blanco, Fernandez-Madrigal, and Gonzalez, 2007) have applied this hybrid method to reconstruct robot's path in a hybrid continuous-discrete state space. Although, the research done so far in mapping is adequate enough for robot navigation to specific targets, but they are devoid of high-level attributes and cognizant capacities being imbued. Now, the trend in robotics is to design agents

[1] Trivia: the word *semantic* is derived from the Greek word *sēmantikos* which means "significant", which in turn derived from the verb *sēmainnein* which stands for "signify" which again stems from the root word *sēma*, that is sign

to operate in human environments close to living beings so the construction of anthropocentric maps endowed with cognizant capacities has become inevitable.

In this chapter, a brief review of different approaches for constructing these maps is proposed. The rudimentary classification of these approaches could be based on the type of environment considered (indoor or outdoor), type of sensors used (laser scanners or RGB-D sensors) or scale of data (single scene or whole environment). The literature review helped to realise the absence of depth sensors usage in indoor semantic mapping even though these sensors have several advantages. In the last section of this chapter, a novel pipeline to address semantic SLAM using depth sensor is proposed.

2.2 Recent Trends in Semantic Mapping

Semantic mapping techniques can be clustered into two different types, depending on where they have been employed, indoor or outdoor. Generally, in many occasions a semantic map is built on top of metric one. The metric map is of course a 3D representation, constructed from many individual scans with different kinds of sensors, as it is very difficult to comprehend from 2D occupancy grids. For indoor case, they can be further distinguished into single-scene and large-scale ones. "The single-scene class gleans those methods that reason about an instance frame with respect to a local coordinate system, also providing conceptual attributes about the observed objects of the scene" (excerpt taken from (Kostavelis and Gasteratos, 2015)). Large-scale approaches on the other hand gradually construct a metric map relative to global coordinate system, and simultaneously annotate it with high-level features (such as walls, doors, floor, ceiling, place labels, object type, etc.). An excellent survey on semantic mapping is done in (Kostavelis and Gasteratos, 2015) and this particular chapter has been inspired by this article.

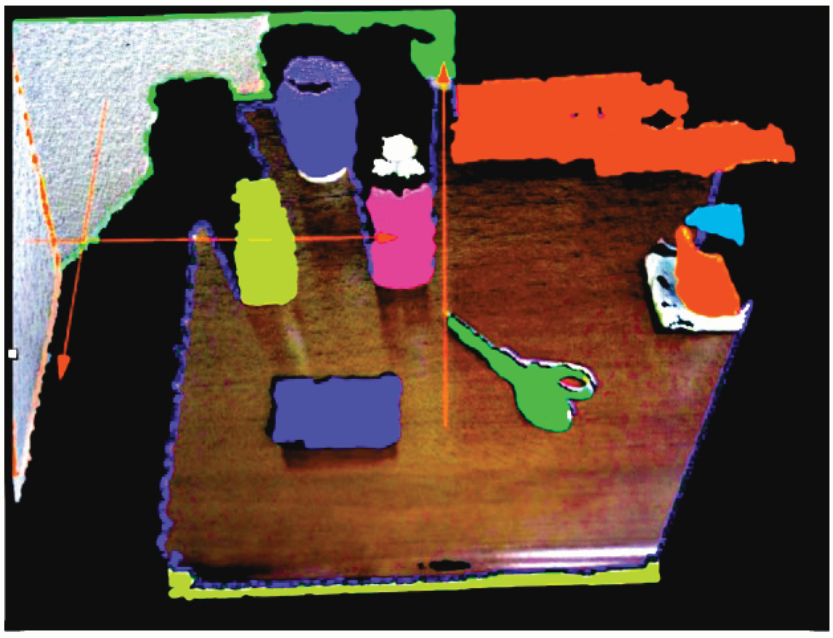


Figure 2.1 – Single scene interpretation. Source: (Trevor et al., 2013).

2.2.1 Indoor Single Scene Interpretation

Nielsen et al. (2004) surmised semantic mapping as an interface between robots and humans. They augmented the metric map with real-world pictures through a single-frame snapshot application (locations are indicated with icons or symbols). Kostavelis et al. (2012) in their

early work used a technique based on SVM (Support Vector Machines) to semantically infer the traversability of a post-disaster environment. Their later work, described in (Kostavelis, Nalpantidis, and Gasteratos, 2012), uses stereo vision and operates on the image plane with the purpose of classifying the traversability of the scene, and they have achieved a remarkable performance for both indoor and outdoor single scene interpretation. Rusu, Gerkey, and Beetz (2008) used multi-sensor (basically a stereo camera and a SICK laser scanner) fusion to help a domestic robot interpret a kitchen scene with several objects. Trevor et al. (2013) recently introduced a single scene point cloud segmentation utilizing connected components practices through RGB-D data. Planar segmentation is performed on the point cloud data to distinguish key components in the scene and then L2 norm based clustering is applied on the color image, in order to detect objects on a tabletop. Swadzba and Wachsmuth (2014) developed spatial 3D feature vectors for single scene classification.

In another single scene interpretation work, Mozos et al. (2012) used a RGB-D camera for visual place classification and Espinace et al. (2013) utilized visual input to interpret object's categories from an exploring robot. The visual input was further treated in a hierarchical fusion manner to characterize the observed scenes in accordance with the existing objects. Bao et al. (2012) used Structure From Motion (SFM) to jointly detect objects and determine the geometry of the scene from two or more uncalibrated images of the scene. They have exploited the correlation between high-level elements (objects) and low-level ones (image features) to extend their previous work (Bao and Savarese, 2011) and coherently solve the SFM and object detection problems. In the most recent work by Cleveland et al. (2017) and Cleveland et al. (2015), a robotic system for generating semantic maps of an inventory in retail environments is developed. Semantic mapping of retail environment, generally, involves labelling of stores where each discrete section of shelving is assigned a department label describing the types of products on the shelf (see Fig. 2.2).

The authors in (Stückler et al., 2015; Choudhary et al., 2014) proposed online semantic parsing (object discovery and object modelling) of indoor environments while simultaneously building the map. While Choudhary et al. (2014) used these objects as landmarks for loop-closures, on the other hand (Stückler et al., 2015) method models geometry, appearance and labelling of surfaces on a RGB-D video. In this thesis, similar to (Cleveland et al., 2017), a particular scene is segmented in the first step using region growing segmentation, and then individual object cluster is recognized based on a novel shape signature; this approach will be called as "*indoor single scene object-level interpretation*" throughout this dissertation. Also, another methodology to interpret large scale scene after global reconstruction is presented in this thesis, which we call as "*indoor large scale higher-level interpretation*".

2.2.2 Indoor Large Scale Interpretation

Major portion of research that has been carried out on semantic mapping has focussed on indoor large scale interpretation in 3D. The more probable reason could be that most of the future service robotic applications involve in working in indoors rather than outdoors. One can easily distinguish the indoor large-scale interpretation methods based on the type of sensor and strategies utilized to construct the metric map.

Accordingly, the works presented in (Nüchter et al., 2005; Blodow et al., 2011; Rusu et al., 2009; Trevor et al., 2010; Rusu et al., 2008) utilize laser scanners to reconstruct the 3D environment. In (Nüchter et al., 2005), 360° map of the scene is captured with SICK laser scanner, and using ICP, a globally consistent map is obtained, the correspondences being established via semantic labels. Similarly, a metric map is developed in (Blodow et al., 2011) using laser scanner and segmentation techniques are applied to generate initial hypotheses about the significance of objects. The authors in (Rusu et al., 2008; Rusu et al., 2009; Trevor et al., 2010) augmented the metric and geometric map with information about the objects present in the scene, while Rusu focussed on kitchen environment and whereas Trevor on detecting horizontal surfaces like

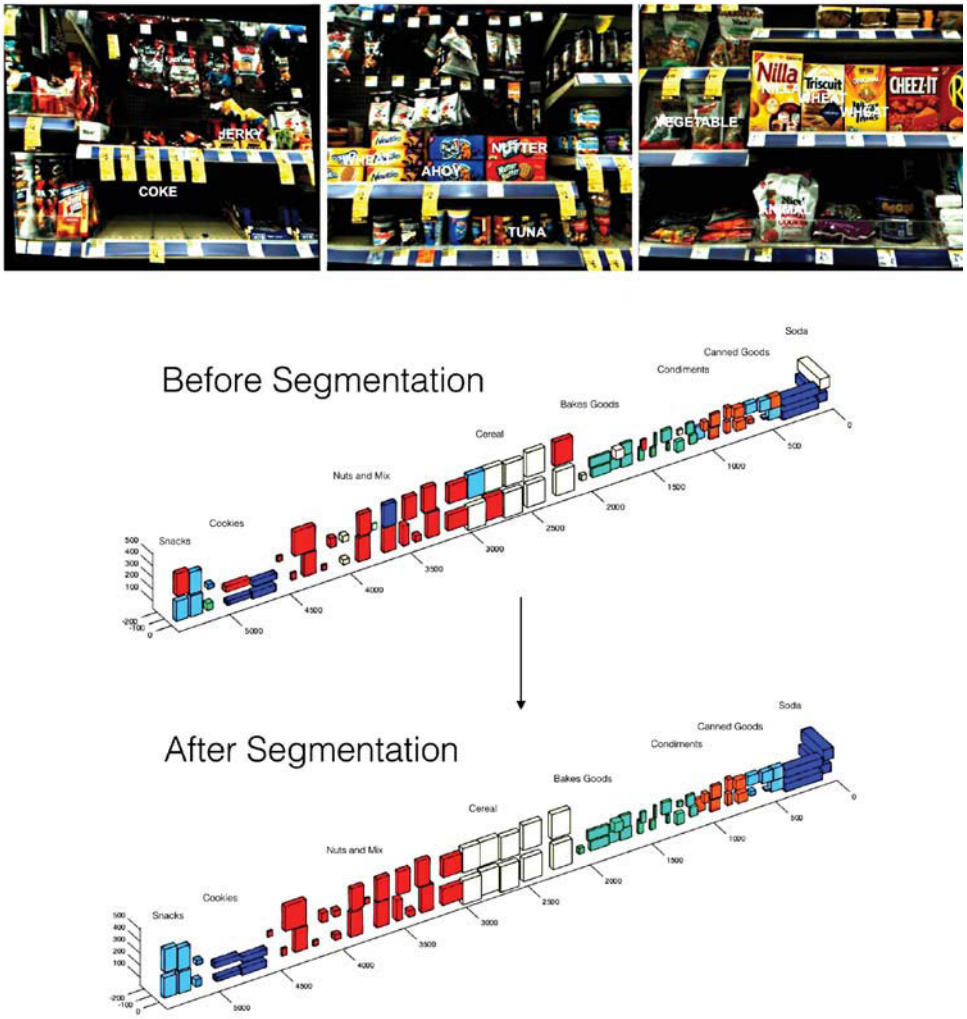


Figure 2.2 – (**Above**) Object detection over store shelf. (**Below**) Automated semantic map generated over an actual Walgreens store aisle using soft object recognition and dynamic programming segmentation. Source: (Cleveland et al., 2015).

tables, shelves, counters etc., in office like environment using Hokuyo UTM-30LX measurements, combined with rotary unit and odometry readings.

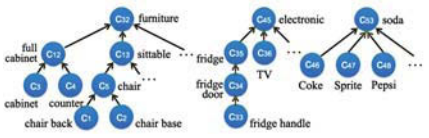


Figure 2.4 – Semantic hierarchy graph. Source: (Wu, Lenz, and Saxena, 2014).

Researchers have also used simple and cheap sensors like Kinect for semantic mapping. Pangercic et al. (2012) used Semantic Object Maps (SOM's) for autonomous service robots performing everyday manipulation tasks in kitchen environments. A PR2 robot acquires the data using an RGBD sensor in a kitchen environment, a SOM$^+$ map is built on this representation from sensor data and queries are performed on this abstraction of SOM (see Fig. 2.3).

In (Kostavelis and Gasteratos, 2013) and (Gunther et al., 2013) a RGB-D sensor, Kinect, has been used to construct globally consistent 3D map using variants of ICP. Kostavelis and Gasteratos developed a two-layer navigational scheme, a 3D SLAM system being at the *lower layer* (numerical navigation), solely based on Kinect data, and a *higher layer* (semantic interpretation) for a spatial abstraction of the input

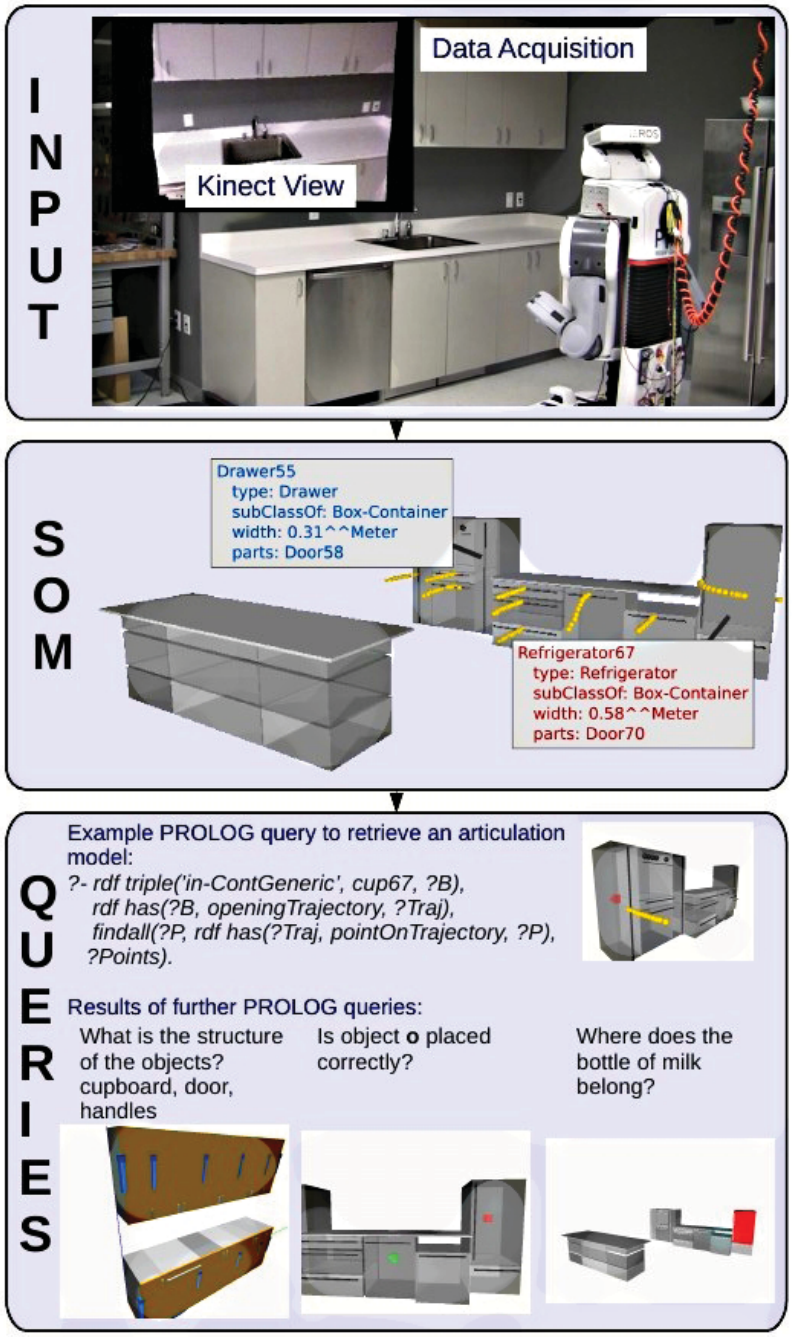


Figure 2.3 – Building of a SOM$^+$ map in a kitchen environment (**top**), SOM+ map representation (**middle**) and a set of robot queries made possible due to such powerful representation (**bottom**). Source: (Pangercic et al., 2012).

space for efficient memorization of the distinct places (e.g., "office", "corridor", etc.). In the numerical navigation layer, SIFT features are detected in consecutive color images, and a point-wise 3D correspondence between the consecutive depth frames of the corresponding SIFT feature points is obtained. These consecutive point clouds are then merged using this feature correspondence information and visual odometry. Later, a refinement step based on ICP alignment of dominant plane (detected by RANSAC) is applied. The *semantic interpretation layer* only pertains to the question of place classification. This has been achieved using *bag of features* technique along with SVM as shown in Figure 2.5. In (Gunther et al., 2013), however, a globally consistent map is developed using SLAM6D toolkit and using Las Vegas Surface Reconstruction Toolkit (LVR) the surfaces are reconstructed and matched with already existing CAD models of furniture and using ICP the poses are adjusted, and then the point clouds are replaced with their CAD models (Trevor et al., 2010).

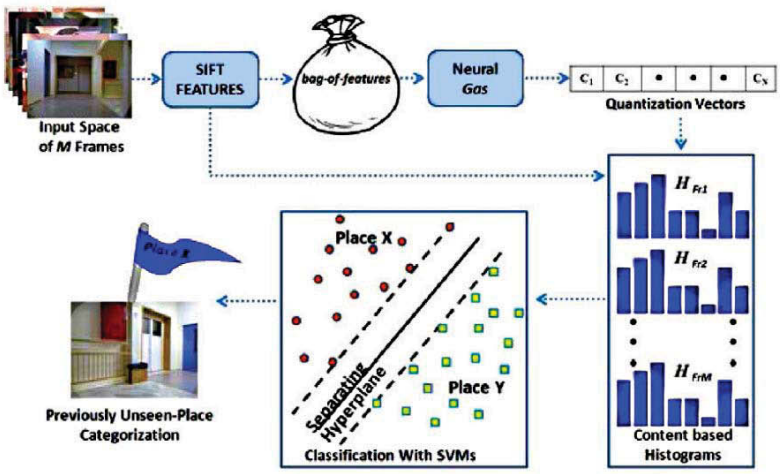


Figure 2.5 – Higher layer for semantic interpretation by Kostavelis and Gasteratos (2013).

Salas-Moreno et al. (2013) proposed a real-time SLAM with hand-held sensors, which harness 3D object recognition to jump over low level geometry processing and produce incrementally built maps directly at the "object oriented" level. As a hand-held depth camera browses a cluttered scene, prior knowledge of the objects likely to be repetitively present, enables real-time 3D recognition and the creation of a simple pose graph map of relative object locations (see Fig. 2.6). Another interesting work on semantic labelling of RGB-D scenes can be found in (Wu, Lenz, and Saxena, 2014). Most of the approaches mentioned are based on *flat labelling* (Ren, Bo, and Fox, 2012; Gupta, Arbeláez, and Malik, 2013) of the scene without considering the important relations between class labels. However, Wu, Lenz, and Saxena (2014) applied *hierarchical labelling*, preserving the relations between class labels, using mixed integer programming to optimize a model isomorphic to a CRF (Conditional Random Field). When labelling with this hierarchy, each pixel belongs to a series of increasingly-general labels; for example, a pixel of class *fridge-handle* would also be of classes *fridge-door*, *fridge* and *electronics* (see Fig. 2.4). The input to the algorithm is a co-registered RGB and depth image pair $\langle I \in \mathscr{R}^{m \times n \times 3}, D \in \mathscr{R}^{m \times n} \rangle$, the goal is to predict the label of each pixel and output the label matrix $L \in C^{m \times n}$, C is the set of possible hierarchical semantic labels. This is achieved by mapping a semantic hierarchy graph (with relations *Is-part-of, Is-type-of*) to the segmentation tree built on the input image. Civera et al. (2011) applied *extended Kalman filter* (EKF) monocular SLAM algorithm in order to create the metric map of the perceived environment and in parallel annotate the scene with semantic labels from an object recognition thread.

An additional class of indoor large scale semantic mapping methods is the one utilizing the laser scanner to form 2D occupancy grids of the environment (see Fig. 2.8). Mozos et al. (2007)

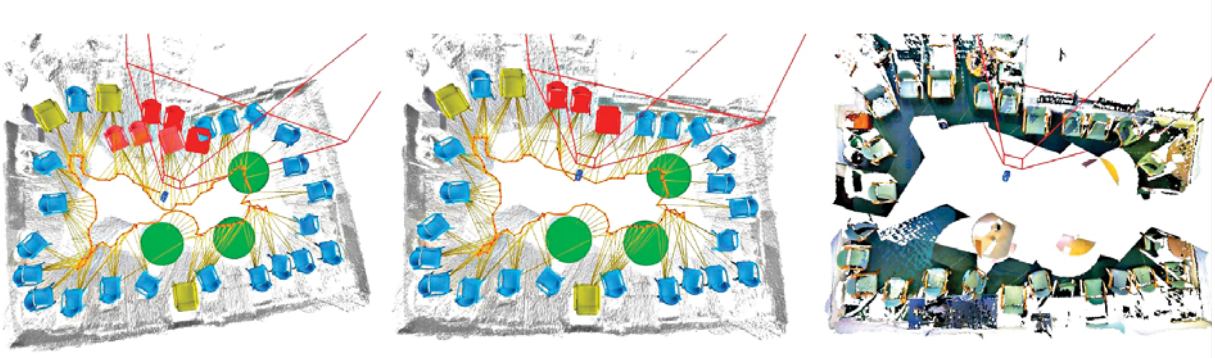


Figure 2.6 – Loop closure. (**left**) Open loop drift during exploration of a room; the corresponding sets of objects are shown in red. The full SLAM++ graph is shown in yellow lines for camera-object constraints and orange lines for camera-camera constraints with orange lines. (**middle**) Imposing the new correspondences and re-optimising the graph closes the loop and yields a more metric map. (**right**) Colored point cloud after loop closure. Source: (Salas-Moreno et al., 2013).

simulated the laser scans from two robots in different maps by using the CARMEN (Montemerlo et al., 2002a) software. The simple features extracted from the range scans are boosted using AdaBoost to achieve a strong classifier. Furthermore, in (Pronobis and Jensfelt, 2012; Pronobis et al., 2010; Ekvall, Jensfelt, and Kragic, 2006; Zender et al., 2008), geometric primitives from laser range scans are extracted and an EKF is applied for the integration of feature measurements. Liu and von Wichert (2014) on the other hand applied standard SLAM and built an occupancy grid map of the environment. Later, he used this map as basis and layered a semantic model upon it. Lawson (Wong, Kaelbling, and Lozano-Perez, 2014; Wong, 2017) developed an approach that combines occupancy grid maps and object-based world models on demand (see Fig. 2.7).

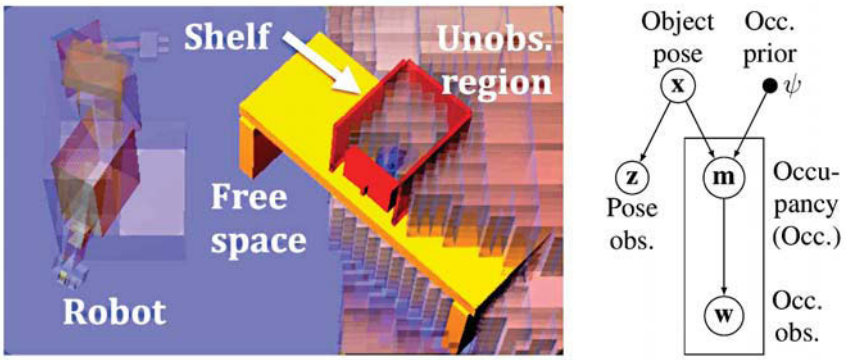


Figure 2.7 – (**Left**) A mobile robot uses object detections to distinguish occupied/free space. And uses free space observations to eliminate possible locations of objects. Source: (Wong, Kaelbling, and Lozano-Perez, 2014). (**Right**) Graphical model for inference across representation models.

Pronobis and Jensfelt (2012) represented semantic information and inference using graphical model, before that, in (Pronobis et al., 2010) he accomplished reasoning with an SVM based cue integration scheme. They also presented a multi-layered semantic mapping algorithm combining multiple visual and geometrical information (Pronobis and Jensfelt, 2011); the metric map is build by exploiting the M-space feature representation. In (Ekvall, Jensfelt, and Kragic, 2006), the generated map was augmented by local and global information about existing objects. Similarly, the method described by Zender et al. (2008) recognizes places and objects by means of laser and visual data, respectively, with the aim to enhance the metric map constructed.

Additionally, the work described in (Krishnan and Krishna, 2010) combines semantic and

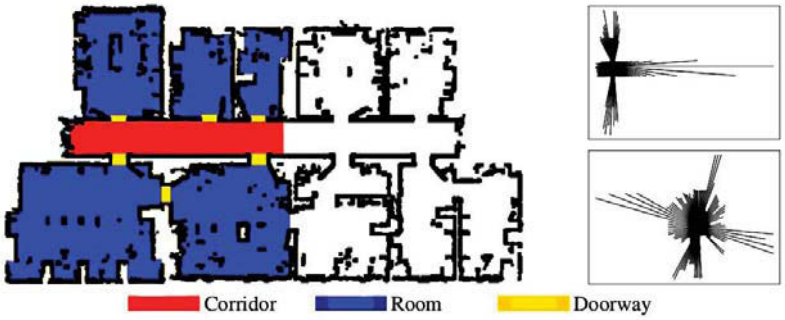


Figure 2.8 – The **right** images depicts example scans recorded in a corridor (**top**) and a room (**bottom**). Source: (Mozos et al., 2007).

topological maps. At the top-level the map is a graph of semantic constructs, where each node is a semantic construct (such as room or corridor) and the edges are the transition regions which connects two semantic constructs (like doorway). Luperto, Quattrini Li, and Amigoni (2014) utilized two laser scanners placed back-to-back to cover 360° around the robot and constructed a metric map which is then later employed during the semantic portioning of the explored area.

A further cluster of indoor large-scale semantic mapping consists of research attempts which exploit stereo vision to acquire depth information of the scene, and use it to solve the SLAM problem. In (Vasudevan et al., 2007), the map generated by SLAM is augmented by the object labels, recognized by means of SIFT features. In (Case et al., 2011), the map is enhanced by including text detection in an office environment. On the other hand, Nieto-Granda et al. (2010) labelled the spatial regions in the map by means of a Gaussian model; the map being constructed using particle filters from ROS (Quigley et al., 2009). Feng et al. (2012) proposed a framework for mobile robot localization in an indoor environment, using concepts like homography and matching borrowed from the context of stereo and content-based image retrieval techniques. The work described by Ranganathan and Lim (2011) utilizes a Visual SLAM system to create a long range metric map (Fig. 2.9), consisting of 3D locations of distinct features observed during robot's perambulation.

It can be observed from the above approaches, that almost all of them have utilized either laser scanners or Kinect like RGB-D sensors or stereo vision for indoor semantic mapping. The time-of-flight depth sensors have hardly been employed for this problem.

2.2.3 Outdoor Interpretation

There have been already multiple approaches proposed to solve the problem of semantic mapping in outdoor environment. While some approaches are very basic, like they calculate the traversability of the path for the robot, which is just a binary classification of the scene operating on the image plane (Kostavelis, Nalpantidis, and Gasteratos, 2012). On the other hand, few approaches achieve complete segmentation and semantic interpretation of the outdoor scene. Bordes et al. (2013) employ multi-sensor fusion and analyse the primitive attributes (e.g., ground, vegetation, structures, obstacles, etc.) of the scene. Sengupta et al. (2013) performed large-scale 3D mapping of the environment and automatically labelled the street scenes using *Conditional Random Fields* (Fig. 2.10). This sophisticated method operates robustly on simple stereo images. In contrast to this holistic approach, Cadena, Dick, and Reid (2015) claim that different perception tasks should be treated as different (software) modules that can be activated or deactivated at will without impairing the rest of the system. The system solves different tasks (geometric reconstruction, semantic segmentation and object detection) in an opportunistic and distributed fashion but still allows communication between modules to improve their respective performances.

Multiclass *Gaussian Process* (GP) classification is adopted by Paul et al. (2012) for semantic

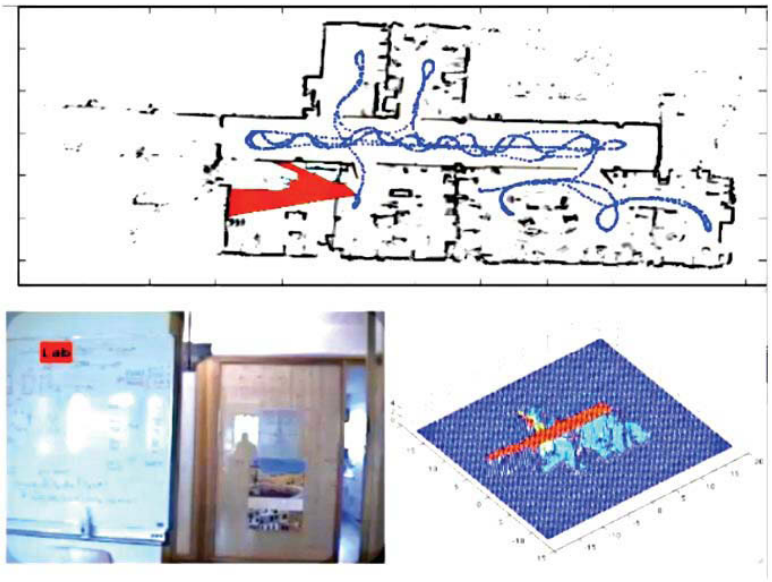


Figure 2.9 – Place labelling in a laser-based map: The top portion shows the current area viewed by the robot (in red) and the robot trajectory from start is shown as a dotted line. The image from the robot camera used for place categorization is shown at bottom left, along with the maximum a posteriori place label from the PLISS algorithm. The label probabilities are accumulated in a grid map of the environment, and the most likely labelled map at the current step is shown at bottom right. Light blue corresponds to the place category "Lab", red to "Corridor", and green to "Copy room". Dark blue corresponds to unknown/unseen areas. Source: (Davison and Murray, 2002; Ranganathan and Lim, 2011).

interpretation of the scene. In the first step, a feature extraction and segmentation is applied on the 3D point cloud and then the feature vectors are fed in a latent kernel classifier function represented by the GP. The uncertainty of the scene objects classified diminishes as the 3D point cloud becomes denser. Steder et al. (2011) developed an approach which uses *bag of words* for loop closure detection and point-feature-based estimations of relative poses to determine a consistent metric map of the environment. This approach has achieved remarkable results to detect reliably previous seen places and calculate accurate transformation between the corresponding scans. Singh and Kosecká (2012) clustered the outdoor scenes into specific regions with their respective labels, utilizing a multi-camera system for long range street scene imagery. The same authors in (Micusik, Kovsecka, and Singh, 2012) performed semantic parsing of street scenes from videos. Saux and Sanfourche (2013) used UAVs to draw semantic inferences from the observations on the ground. They used an online gradient boost algorithm to interactively interpret context dependent detectors. Katsura et al. (2003) proposed a weather-invariant vision based outdoor navigation method endowed with object recognition attributes. They also proposed a comparison method in which the robot firstly recognizes objects in images and then compare recognition results of learned and target images.

2.3 Proposed Approach

As mentioned before, depth sensors are not exploited enough for indoor semantic mapping even though they have several advantages (will be discussed in Chapter 3). In this thesis, a single time-of-flight depth sensor (SwissRanger SR4000) is used to reconstruct the indoor environment and interpret the scene. Similar approach is followed as that of (Nüchter and Hertzberg, 2008), except localize the objects in the global map (see Fig. 2.12). The point cloud generated from the SwissRangers is extremely noisy to detect the objects in the reconstructed global point cloud.

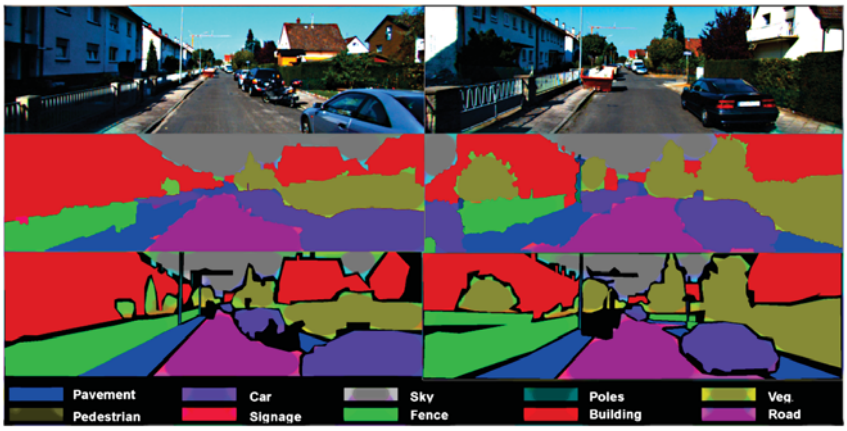


Figure 2.10 – *Semantic image segmentation:* The **top** row shows the input street-level images and the **middle** row shows the output of the CRF labeller. The **bottom** represents the ground truth. Source: (Sengupta et al., 2013).

However, individual scans can be verified for the presence of the objects and then localize them. Our approach (see Fig. 2.11) deals with two different levels of interpretation: object level and generic surfaces. The object level interpretation is performed on a single scene while the higher-level interpretation is done on the global map. To the best of our knowledge, this methodology has never been tried and moreover using time-of-flight camera like SwissRanger.

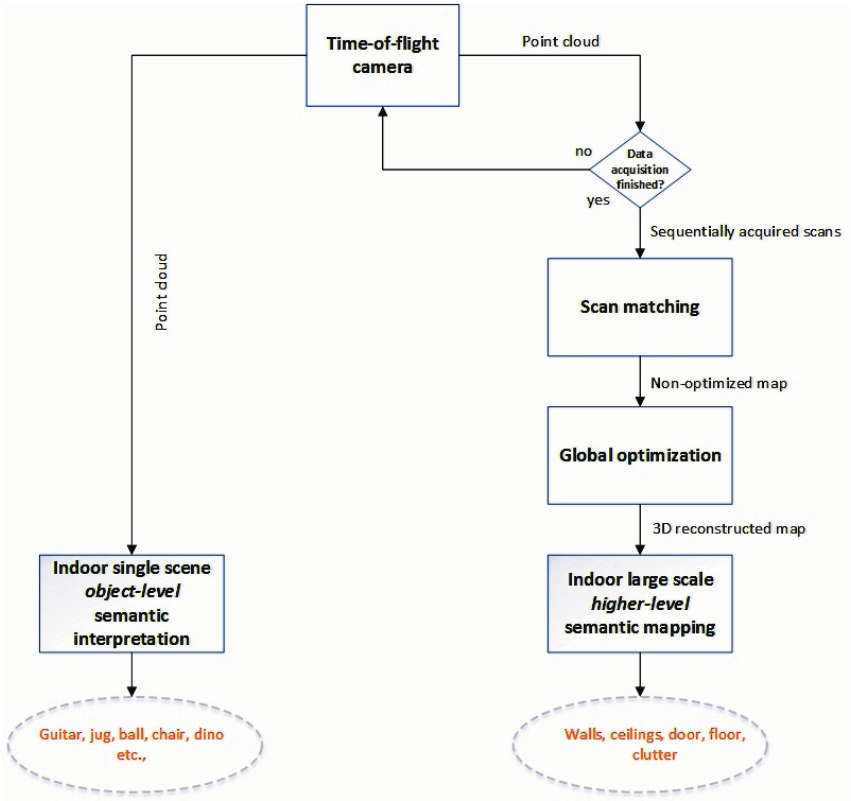


Figure 2.11 – Flowchart of the semantic mapping process. The approach is very similar to that of Nüchter and Hertzberg (2008), where they semantic label higher-level planar surfaces using some rules and detect objects using trained classifier.

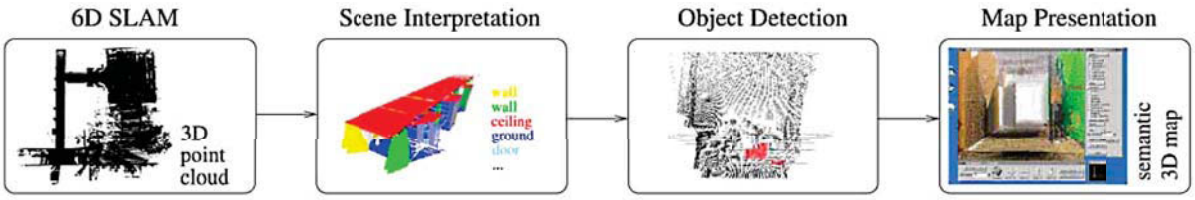


Figure 2.12 – Semantic mapping pipeline. Source: (Nüchter and Hertzberg, 2008).

Chapter

3

Time-Of-Flight Camera: Working Principle and Noise Filtering

Before the year 2000, stereo cameras have been used to obtain 3D data of the environment. However, stereo cameras are not good choice as they do not provide 3D information for textureless surfaces. In the recent years, a new generation of active cameras based on the Time-of-Flight principle (ToF) has been developed. They work with an active illumination and generate 3D data at video frame rate. Along with several advantages, they also have few shortcomings. This chapter discusses different kinds of noise present in the data obtained from ToF camera and also give brief introduction about its working principle. SwissRanger camera like any other ToF camera have systematic and non-systematic errors. Systematic errors are predictable and can often be removed by calibration, whereas non-systematic are unpredictable and removed by applying filters. Jump edges are the most prominent and popular non-systematic errors present in this sensor data. It is very important to remove the noise as it can affect every stage of robotic applications. A new method to filter jump edges in the range images produced from ToF camera is described, implemented and evaluated in this chapter. Jump edges are seen as smooth irregular sigmoid shape or curved transition between two overlapping surfaces separated by some distance. The new filter's efficiency is compared with a state-of-the-art method. The comparison is based on the quality of filtered image, computation time for filtering and also its impact on registration of successive scans and reconstruction of the whole scene.

3.1 Introduction

Over the last decade, substantial and continuous improvement in microelectronics, micro-optics and micro technology for robotics oriented 3D-sensing, has led to invention of Time-of-Flight (ToF) cameras to capture depth images. Since their development, they have surpassed and outperformed the past technology in range imaging. Further efforts are put to optimise and design compact and more efficient prototypes. Many commercial sensors are now available at affordable prices as shown in Figure 3.3. ToF camera delivers 3D imaging at a high frame rate, simultaneously providing reflectance data and range information for every pixel. Depth-intensity pixel-associated images at a video frame rate without any need of any extra mobile components with additional technical advantages, as mentioned in (Foix, Alenya, and Torras, 2011b), such as "robustness to illumination changes" and low weight, make it foreseeable that ToF camera will replace previous solutions, or, alternatively, complement other technologies, in many areas of application. However, with all these excellent attributes, ToF cameras suffer from various errors. Larger fluctuations in precision due to external interfering factors (e.g., sunlight, other source of illumination in the environment), distance orientations, object reflectivity, motion blur makes ToF cameras unsuitable for most robotic application without calibration and filtering.

3.1.1 ToF Range Imaging Principle SwissRanger SR4000

Range imaging is blend of two different technologies *viz.*, distance measurement and imaging. A NIR (near infrared = 850 nm) modulated wave (of frequency f) at few tens of MHz is directed into the scene, the CCD/CMOS sensor detects the reflected IR and measures the phase of the returned signal at each pixel, as shown in Figure 3.1. Every pixel on the sensor samples the amount of light reflected by the scene four times at equal intervals for every period (m_0, m_1, m_2 and m_3). The phase φ, offset B, amplitude A and depth D are given as (Foix, Alenya, and Torras, 2011b):

$$\varphi = \arctan\left(\frac{m_3 - m_1}{m_0 - m_2}\right) \tag{3.1}$$

$$B = \frac{m_0 + m_1 + m_2 + m_3}{4} \tag{3.2}$$

$$A = \frac{\sqrt{\left[m_3 - m_1\right]^2 + \left[m_0 - m_2\right]^2}}{2} \tag{3.3}$$

$$D = L\frac{\Delta\varphi}{2\pi} \tag{3.4}$$

where $L = \frac{c}{2f}$ is the *ambiguity-free distance range* of the sensor.

Range imaging combines distance measurement technology with the advantages of those of imaging arrays. Simplified, it just enables each pixel to measure the distance towards the corresponding object point. This is regarded as an array of range finders, hence they are called as smart pixels. Figure 3.2 presents the principle of range imaging. The measured distances in connection with the geometrical camera relationships can be afterwards used to compute the 3D coordinates which represent a reconstruction of the imaged object/scene (Kahlmann, Remondino, and Ingensand, 2006). For SwissRanger ToF cameras, the absolute origin is at the center of the optical filter, *i.e.*, at the intersection of the optical axis with the front face of the camera (see Fig. 3.7) and the depth is given/calculated from this origin, unlike the perspective camera projection model where the origin lies at the optical center.

3.1.2 Depth Measurement Errors Classification

The performance of distance measurements by ToF is affected by number of errors. These errors can be broadly classified as *systematic* and *non-systematic*. While the former are predictable and

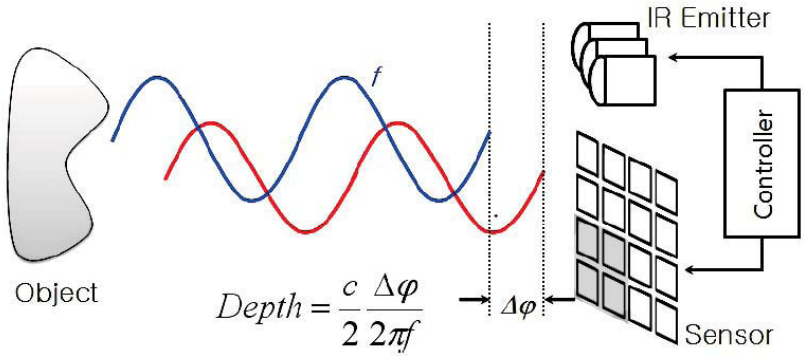


Figure 3.1 – The principle of ToF depth-camera. Source: (Kolb et al., 2010; Kang et al., 2011; Lee, Choi, and Horaud, 2013; Foix, Alenya, and Torras, 2011b).

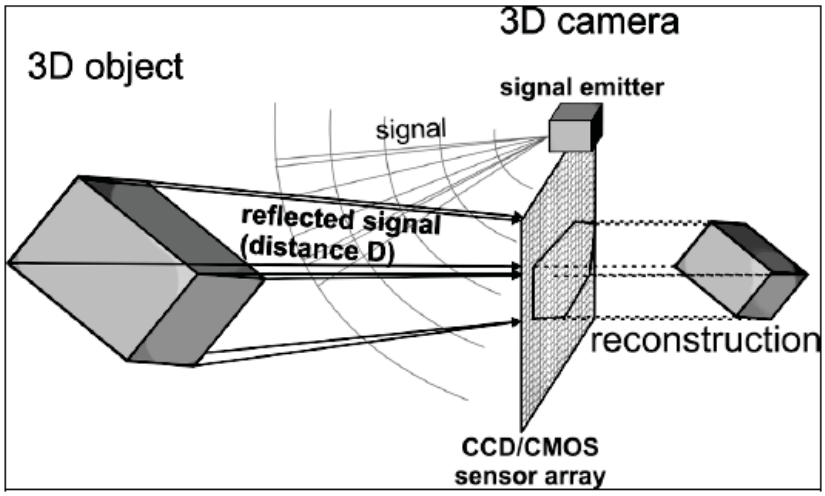


Figure 3.2 – ToF Range imaging principle. Source: (Kahlmann, Remondino, and Ingensand, 2006).

correctable by calibration but the latter cannot be predicted and generally removed by filtering.

▷ **Non-systematic errors**. Signal-to-noise ratio (SNR), inter-reflection, light scattering, motion blurring errors are non-systematic in nature, as they depend on the scene configuration and cannot be predicted.

- **Signal-to-noise ratio (SNR)**. SNR distortion appears in scenes which are not uniformly illuminated, low illuminated areas being amenable. SNR is highly dependent on the amplitude, the IT (Integration Time) parametrization and the depth uniformity of the scene.
- **Multiple ways reflection**. They occur due to the interference of multiple light reflections captured at each sensor's pixel. Due to this, hollows and corners appear rounded off and occluding shapes have a smooth transition.
- **Light scattering**. Light scattering is another non-systematic error which cannot be predicted as the topology of the observed scene is unknown a priori. This arise due to multiple light reflections between the camera lens and its sensor. This leads to depth underestimation over the affected pixels, because of the energy gain produced by its neighboring pixel reflections. These errors are pertinent when the objects are close to the sensor. The closer the object, the higher the interference.

Figure 3.3 – Commercially available Time-of-Flight cameras. Source: (Schaller, 2011).

- **Motion blurring**. They normally occur when the sensors are used in dynamic environment, due to physical motion of the object or sensor during the integration time used for sampling.

▷ **Systematic errors**. Wiggling (circular error), integration time (IT) related, built-in pixel related, amplitude and temperature related errors are predictable and are systematic.

- **Wiggling or Circular error**. Depth distortion occurs due to irregularities in modulation process, as a result the emitted infra-red light is not sinusoidal. This error produces an offset that depends only on the measured depth for each pixel.
- **Integration time (IT)-related error**. Integration time is sometimes synonymously used for *exposure time*, but it is actually the time interval during which the camera's clocks are set to trap and retain charge. IT can be selected by the user, it has been observed that for the same scene with different IT causes different depth values in the entire scene.
- **Built-in pixel-related errors**. They arise either due to different material properties in CMOS-gates or capacitor charge time delay during the signal correlation process. The former results in constant pixel-related distance offset, leading to different depths measured in two neighbor pixels corresponding to the same real depth. While the later results in latency-related offset errors and observed as a rotation of the image plane, *i.e.*, a perpendicular flat surface is viewed with wrong orientation.
- **Amplitude-related errors**. They occur due to low or overexposed reflected amplitudes. Depth accuracy depends on amount of incident light. The higher the reflected amplitudes, greater is the depth accuracy. Low amplitude appears more often in the border of the image as the emitted light power is lower than in the center, leading to awry depths. On the other hand if the object is too close or if the integration time is chosen too high, saturation can appear and depth measurements will not be valid. These errors are caused mainly by:
  - Systematic non-uniform NIR LEDs illumination causes depth misreadings at pixels distant from image center.
  - Low illumination for scenes with objects at different distances.
  - Difference in object reflectivity cause different depth measurements for pixels at the same constant distance.
- **Temperature-related errors**. Internal camera temperature affects depth processing, depth values suffer from a drift in the whole image until the temperature of the camera is stabilized. This is due to the fact that semiconductor materials are sensitive to changes in temperature. Generally an over-estimation in measured distances is found when the sensor started working and operating at high temperatures.

A detailed description of these errors can be found in (Foix, Alenya, and Torras, 2011b). In this chapter, we deal with one of the most prevalent non-systematic error which are the result

of inter-reflection, called jump edges. Inter-reflections also called as multiple-ways reflection, occurs due to occlusions in concave objects, e.g., corners or hollows and edges. In this case, the signal can take multiple ways through reflection before returning to the receiver, and the re-emitted signal is superposition of illuminated light that has travelled a different distance, this phenomenon is called as *multimodal reflection*. As a result of this, hollows and corners appear rounded off and occluding shapes connected with a smooth transition. The reason for this, as mentioned in (May et al., 2009b), is that, it happens as a consequence of diverging measurement volume.

3.1.3 Jump Edges

Jump edges appear as irregular sigmoid shape or curve shaped transition between two surfaces which are at different distances, or between foreground object and background, as shown in Figure 3.5. The jump edges keep changing as the field of view is changed (see Fig. 3.4), due to this, the registration between two scans is lousy and hence the whole reconstruction is of very poor quality. These errors were successfully removed by (Fuchs and May, 2008) to the most extent by applying threshold on the opposing angles formed by the focal point and two neighbour data points (this method is referred as Angle Method or AM throughout this chapter). However, this method theoretically fails if the angle between two planes is greater than the threshold and hence fails to remove some jump edges, which could lead to bad registration or increase in the convergence time of applied registration algorithm. Hence, a novel method based on Line-of-Sight is proposed, implemented and compared with (Fuchs and May, 2008); the approach is detailed in Section 3.4.

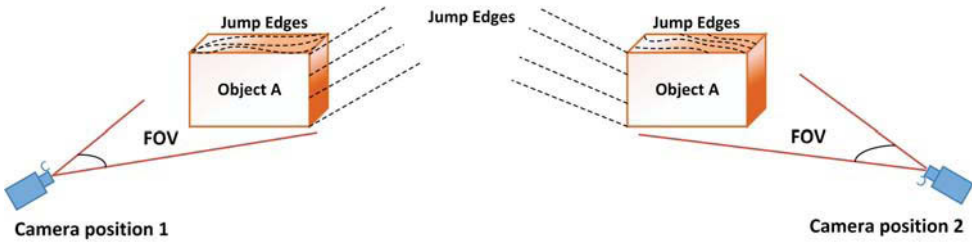


Figure 3.4 – Jump edges depend on the field of view, and can thus lead to bad scan registration. Source: (May et al., 2009b).

3.2 Related Work

Several methods have been proposed to overcome the identification and/or correction of jump edges (May et al., 2009b). Pathak, Birk, and Poppinga (2008) proposed a Gaussian analysis for correcting multi-modal measurements. The main drawback of this method is the computation time for the Gaussian fitting and integration over 100 images for each frame; which significantly also reduces the frame rate. Sappa, Restrepo-Specht, and Devy (2001) presented an approach that is aimed at identifying and classifying edges. It uses the fitting of polynomial terms to approximate scan lines. These scan lines are connected at edge points. The strength of this approach is that it also performs a classification of edges into jump edges and crease edges; "Crease edges are those points in which a discontinuity in the surface orientation appears", e.g., in corners or hollows. In (May et al., 2009b) and (Fuchs and May, 2008), the authors proposed a simplistic method to remove this errors. From a set of 3D points $P = \{p_i \in \mathbb{R}^3 | i = 1, \ldots, N_p\}$, jump edges J can be selected by comparing the opposing angles $\theta_{i,n}$ of the triangle spanned by

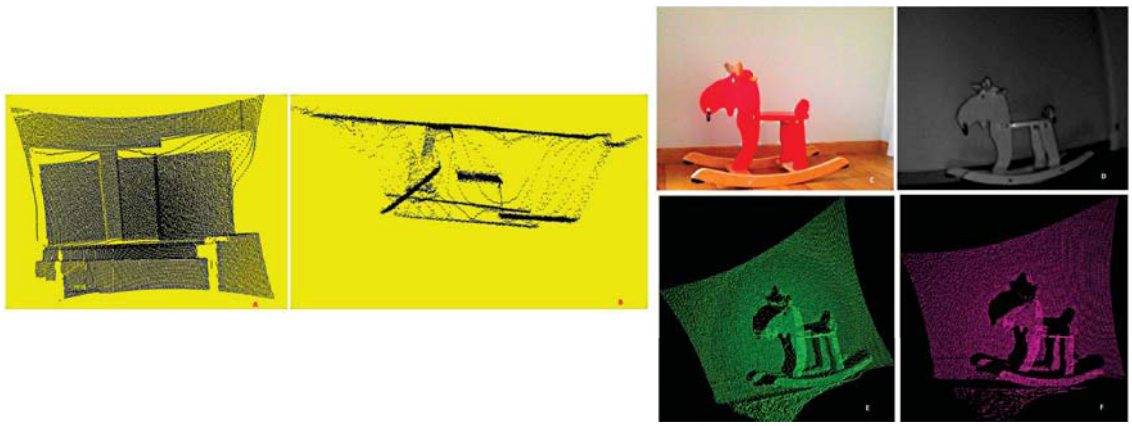


Figure 3.5 – Range image of three planar surfaces from SR-4000 camera. **A**: Jump edges can be seen as curved transition from the front surface to the background. **B**:-Top view showing jump edges between the planar surface and the background. **C**:- RGB test image. **D**:-Amplitude Image. **E**:- Unfiltered Point Cloud, Jump edges can be seen connecting object to the background. **F**:- LoS filtered Point Cloud with jump edges removed (see Section 3.4).

the focal point $f = 0$, point p_i and its neighbours with a threshold θ_{th}. However, this method will remove valid points if the inter-planar angle value is greater than θ_{th}. We have proposed a method which is able to deal with this situation and yet remove the jump edges.

3.3 SwissRanger Depth Camera

SwissRanger time-of-flight cameras are manufactured by a Swiss Company: MESA Imaging[1]; since the year 2009, different camera designs (SR3000, SR4000, SR4500) with unique features commercially exist in the market. Both flavours of SR4000 (5 m and 10 m maximum range) are used in this thesis (see Fig. 3.6). The specifications of SR4k are shown in Table. 3.1, they are the same for both 5 m and 10 m versions, but they only differ in Modulation Frequency (MF) and Absolute Accuracy. Before the acquisition of data, the Integration Time (IT) and MF are set to the optimum values depending on the illumination present in the environment.

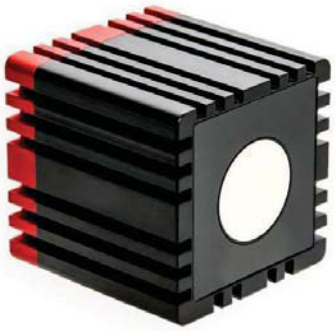


Figure 3.6 – The SR4k time-of-flight camera. Both 5 m and 10 m versions have similar design. Source: (*SwissRanger, Heptagon*).

Parameter	Value
Illumination Wavelength	850 nm
Maximum Frame Rate	50 fps
Pixel Array Size	QCIF 176 (h) × 144 (v)
Field of View	43.6° (h) × 34.6° (v)
Pixel Pitch	40µm
Voltage	12.0 V
Angular Resolution	0.24° × 0.39°
Operating Temperature	+10 ° Celsius to +50 ° Celsius

Table 3.1 – SR4k depth camera specifications (please refer Appendix B for complete specifications)

3.4 Approach

3.4.1 Line-of-Sight-based Jump Edge Filtering:

From the working principle of ToF depth estimation (see Fig. 3.2), every region/point in the scene is illuminated by a near infra-red light, and the system measures the round trip time to bounce the point in the scene and reflect back to the receiver. In principle, the infra-red light cannot illuminate any point behind and beyond another point which is currently illuminated, as the current point blocks further progression of light, and the signal is reflected back to the sensor. So, it is not possible to have two points on the same line-of-sight. Moreover, due to diverging measurement volume (May et al., 2009b), as each pixel has a field of view (around 0.290° × 0.340° for SR-4000), the depth measurement technique will look like a series of non-intersecting cones arising from the *smart pixels* and limelight each point in the scene, and estimate the depth. As a result, each *smart pixel* has discrete point in the scene being "cynosure". But the jump edges seems to be connecting two surfaces/objects at different distances/depths, and they lie on the pixel's field of view of the points in the foreground points. These jump edge points are present inside the pixel's field of view (FOV) of foreground point but with different depth. So, a filtering method based on limelighting each point in the scene has been proposed. We also remove the random noise to some extent by removing the points with no neighbors.

From the point cloud $\mathbf{P}$, the depth image $\mathbf{D}[m_{\times}n]$ is calculated by projecting the 3D points on a 2D image such that the pixel $\mathbf{D}[\mathbf{k},\mathbf{l}]$ has depth information of corresponding point in $\mathbf{P}$. Given a point $\mathbf{P_i}$ in the cloud $\mathbf{P}$, with depth $\mathbf{D_i}$, the eight neighbours of it are checked in $\mathbf{D}$ to see if they lie on the Line-of-Sight of $\mathbf{P_i}$ (see Fig. 3.7). All the neighbours which lie on this line are marked as jump edges and removed (see pseudo-code 1). Furthermore, a metric (Fig. 3.8) is defined, which removes the points which lie on the small spherical segment. The height (ε) of the spherical cap follows an inverse relation with the distance ($\mathbf{d_{i,n}}$) between the point $\mathbf{P_i}$ and its eight neighbours ($\mathbf{P_{i,n}}$). However, for experiments constant values ($\varepsilon' = 0.1$ mm) are taken if $\mathbf{d_{i,n}} > 0.1$ mm, as it results in effectively retaining valid neighbours. A relation between width of the pixel's FOV, $\mathbf{d_{i,n}}$ and ε at a point in the scene need to be established. This is because of the diverging pixel's field of view (0.290° × 0.340° for SR-4000 5 m measurement range camera and varies with camera make-up). The ε metric is similar to the mentioned *limelighting principle*, and it gives every point in the scene its importance without sharing with neighbour points.

[1] As of the year 2014 the MESA Imaging company has been acquired by Heptagon.

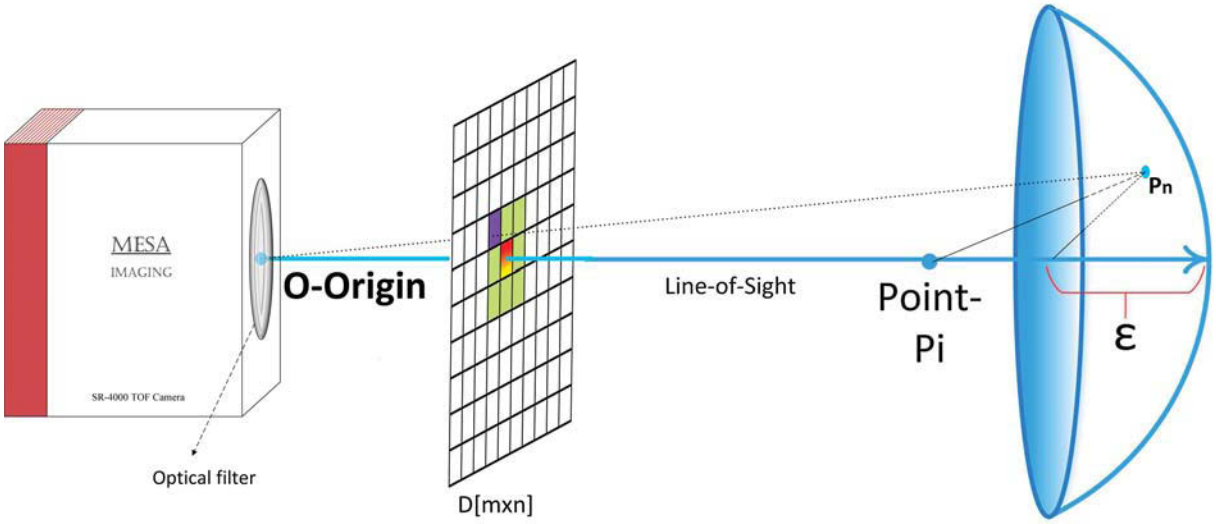


Figure 3.7 – Line-of-Sight method principle: here the point $\mathbf{P_i}$ is considered, and its eight neighbours in the 2D range image are checked if they stand on the line-of-sight, originating from the absolute origin of the camera passing through $\mathbf{P_i}$.

Algorithm 1 Line-of-Sight pseudo-code for range image filtering

Inputs:
In_Cloud: Depth image **D**[m×n] $\left(\mathbf{D}[\mathbf{i},\mathbf{j}] \in \mathbb{R} | \mathbf{i} = 1 \cdots m,\ \mathbf{j} = 1 \cdots n\right)$
Initialize:
Orig_Cloud, *Curr_Cloud* ← *In_Cloud*
for i = 1 to *all pixels in Orig_Cloud* **do**
  for j= 1 to *all points in 8 neighbourhood of Orig_Cloud[i]* **do**
    if depth$\left(Orig\_Cloud[i,j]\right) \geq$ **depth**$\left(Orig\_Cloud[i]\right)$ **then**
      if *Orig_Cloud*$[i,j]$ *is on spherical cap defined by* ε **then**
        Remove *Curr_Cloud*$[i,j]$
    else
      if *Orig_Cloud*$[i]$*is on spherical cap defined by* ε **then**
        Remove *Curr_Cloud*$[i]$
return *Curr_Cloud*: **Filtered Point Cloud's 2D image**

3.5 Results, Evaluation and Discussions

3.5.1 Datasets

Experiments are carried out basically on three different types of scenes. The first scene has three planes slightly inclined to each other and positioned at different distance from camera. The second scene has **Z** shaped custom designed plane. And the last dataset is made with two planes having very high or low inter-planar angle.

3.5.2 Results

From Figure 3.9, it can be observed that LoS-filtered image is more close to ground truth than Angle Method (AM) (Fuchs and May, 2008). The ground truth is made with an image editor, manually removing the jump edges and retaining only surfaces. Ellipses have been drawn on jump edge points which are not removed. AM failed at removing eleven locations whereas LoS at only one position. In Figure 3.10, the top series of images (A, B, C, D, E) elaborates the comparison, visualizing the removed points (green). The bottom series of images correspond

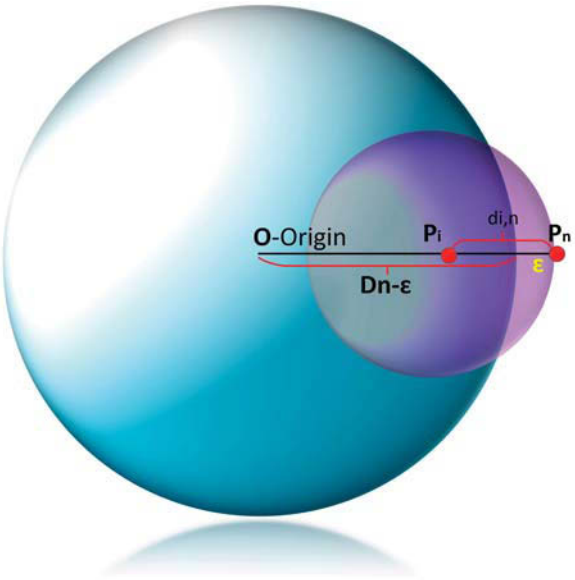


Figure 3.8 – Spherical cap is formed by intersection of two hollow spheres, with radii $(\mathbf{D_n} - \varepsilon)$ and $\mathbf{d_{i,n}}$. The eight neighbors of $\mathbf{P_i}$ (e.g., here $\mathbf{P_n}$) are checked if they lie on the spherical cap and are removed as jump edges if they lie. ε defines the size (height) of the spherical cap, and follows inverse relation with $\mathbf{d_{i,n}}$, due to diverging pixel's FOV (refer text).

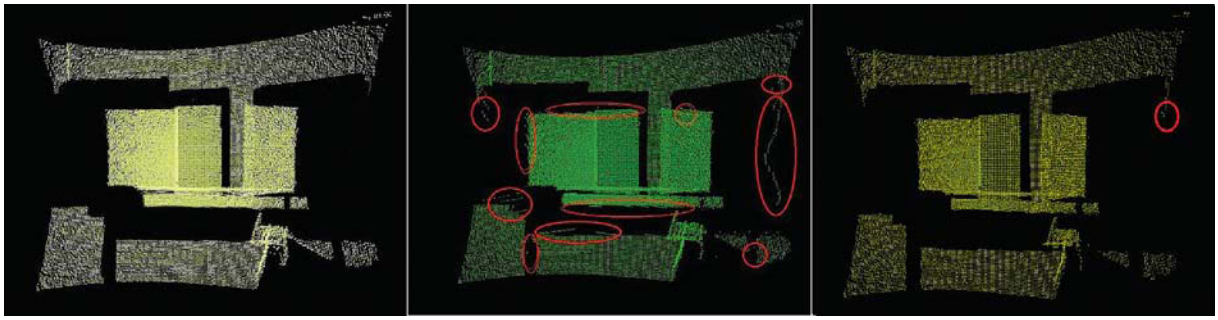


Figure 3.9 – Comparison with ground truth: **Left**- Ground truth. **Middle**- From Angle Method (AM). **Right**- From LoS method. Ellipses show the points not removed (11 for Angle, 1 for LoS).

to second scene dataset with **Z** shape (top-view). In this case also LoS performs much better at removing points inside the pocket of **Z**. The third dataset is made to test methodological failure of AM. It fails when the angle between two planes is more than the threshold applied. It can be seen in Figure 3.11, AM removes the valid points on the V junction, whereas LoS has retained those points. Both the algorithms are applied on the median filtered range images, as suggested in (Fuchs and May, 2008).

Experiments are also conducted to test the effect of filtering on the task of registering two scans. PCL ICP (Rusu and Cousins, 2011) has been used to evaluate the performance. Here, three different pairs of scans (with two scans having 2.5°, 5.0° and 7.5° of angular rotation between them without translation) are registered. Both the filtering methods are compared based on ICP fitness score (sum of squared Euclidean distances between corresponding points in the source scan and the target scan). As observed in Table 3.2 and Table 3.3, LoS has lower fitness score and thus better registration. And also LoS is faster than AM to implement. The average computation time for filtering single range image with LoS method is 47.19 ms and for AM is 65.18 ms on a Hewlett-Packard ZBook installed with Linux (Ubuntu 14.04) platform, programmed using C++. Each range image has 25344 [144 $\times$ 176] points.

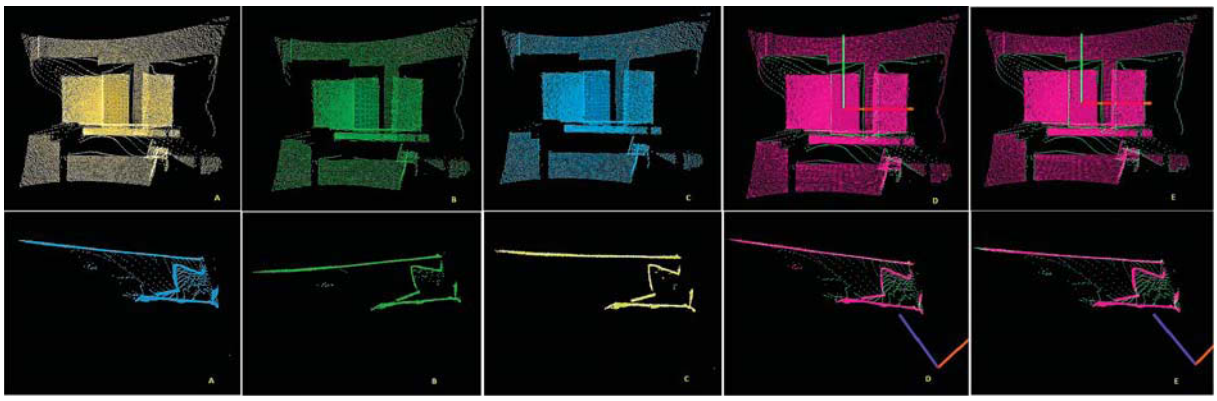


Figure 3.10 – Effect of filtering on two scenes with different objects (planes): Top and Bottom (top view of z-shaped object): **A**- Unfiltered depth image, **B**- Filtered with AM, **C**- Filtered with LoS, **D**- Filtered by AM (Green- removed points, Purple- remaining points), **E**- Filtered by LoS method (Green- removed points, Purple- remaining points).

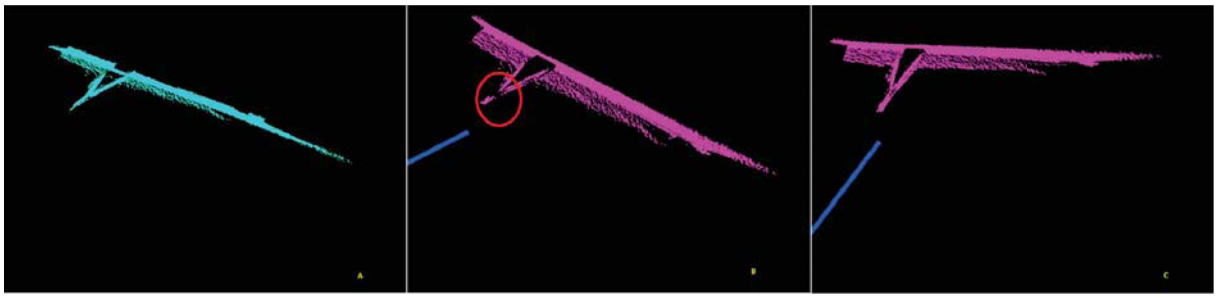


Figure 3.11 – Experiment on planes with inter-planar angle more than threshold: **A**- Unfiltered image, **B**- From AM, **C**- From LoS method. Clearly, points are removed at the cross-section of planes due to thresholding on angle for Angle Method, whereas the points are retained for LoS.

3.5.3 Globally Consistent 3D Scene Reconstruction

Using 3DTK (Borrmann et al., 2008a; Nüchter et al., 2007; Borrmann et al., 2008b), globally consistent 3D reconstruction of an indoor scene is done from the LoS filtered images. It can be seen (Fig. 3.12) that the 3D reconstruction is very neatly done, even the door handle is perfectly represented in the model.

3.6 Conclusion

Jump edges drastically lead to bad registration, as false points form depending on the field of view. It is very important to remove these wrong points before registration or mapping. In this work, a novel method to remove jump edges is presented. Our results are also compared to the method which is most often used. The proposed method is able to perform much better than AM in terms of quality of filtered image, computation time to apply filter and also on registration of scans. We have also precisely reconstructed a complete indoor office environment with LoS filtered images. Based on these qualitative and quantitative results, our method outperforms AM and is better suitable to filter non-systematic noise in SwissRanger camera in particular and ToF camera in general.

In the next chapter, the algorithm to match two noise free scans is presented. It also discusses other state-of-the-art invariants applied.

$\varepsilon(m)$	$\varepsilon'(m)$	Computation Time (s)			ICP Fitness-score $\times 10^{-4}$		
		2.5°	5.0°	7.5°	2.5°	5.0°	7.5°
0.001	10^{-4}	11.5	12.1	12.3	1.2885	4.7974	33.5265
0.002	10^{-4}	15.9	16.6	16.7	1.2222	4.2222	30.9613
0.003	10^{-4}	28.5	28.5	27.6	1.1101	3.6782	27.7957
0.004	10^{-4}	57.4	53.6	53.7	0.9356	3.0296	23.0192

Table 3.2 – ICP Fitness-score: LoS method

Parameter	Computation Time (s)			ICP Fitness-score $\times 10^{-4}$		
Theta $(\theta°)$	2.5°	5.0°	7.5°	2.5°	5.0°	7.5°
190	9.60	9.61	10.30	1.5359	5.8441	37.117
180	9.63	9.65	10.32	1.5359	5.8441	37.117
170	10.6	10.85	11.45	1.3615	5.077	35.215
160	11.1	11.50	12.30	13.128	4.8052	33.182

Table 3.3 – ICP Fitness-score: AM

Rotation in degrees	Computation Time (s)	ICP Fitness-score $\times 10^{-4}$
2.5	9.275	1.536
5.0	9.565	5.844
7.5	10.226	37.117

Table 3.4 – ICP Fitness-score: Unfiltered data

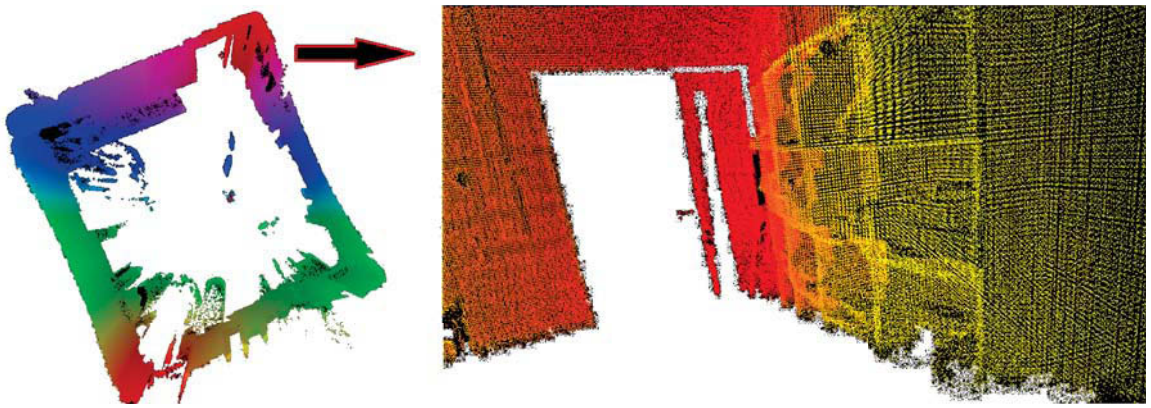


Figure 3.12 – Globally consistent 3D reconstruction of an indoor scene. **Left**- Top view of the scene. **Right**- Part of the 3D scene, a closed glass door (appears transparent) with a door handle, precisely reconstructed, and multi-cupboard armoire on the right side.

Chapter

4

3D Scan Matching with ICP

The problem of 3D scan matching is of utmost importance for the construction of metric representation of the environment, for localization and navigation planning in 3D space, for object recognition and manipulation. 3D scan registration can be formulated as the problem of finding the relative transformation between two 3D point clouds that best aligns them. Since, almost from four decades, several methods have been developed to solve this problem; originally, it being more associated with biomedical image registration, but now faced in almost all robotic applications. In this chapter, a brief history of scan matching problem is presented and, different approaches developed over the years to solve it are discussed. The most applied method: Iterative Closest Point (ICP) algorithm, is presented in detail and several other variants which evolved from it are also discussed briefly. ICP has been used extensively in this thesis for scan matching in SLAM and for designing a novel shape descriptor; the results are presented at the end of this chapter.

4.1 Introduction

With the advent of inexpensive depth sensing devices in recent years; the research in robotics, computer vision and ambient application technology involving 2D imaging and LIDAR (Laser Imaging Detection And Ranging) scanning has shifted towards real-time reconstruction of the environment based on 3D point cloud data. Point clouds either structured (generated from structured light based sensors such as Microsoft Kinect and Asus Xtion) or unstructured (from time-of-flight sensors like SwissRanger, Softkinetic Depthsense, etc.) can be directly used to detect and recognize objects in the environment where ambient technology is used or can be integrated over time to completely reconstruct a 3D map of the camera's surroundings. Each point in the point cloud corresponds to a point in the physical world at a distance equal/proportional to half the distance travelled by the light from the emitter back to the receiver after being reflected by the point. Most mobile robotic applications use sensors whose framework is based on this principle or Triangulation techniques. The point cloud data-structure holds the 3D coordinates of each point in the simplest representation. These coordinates are relative to scanning device coordinate system. In order to reconstruct a complete model or environment from different scans taken at different angles or time intervals, one has to move all the points in each scan to the same coordinate system. The alignment of these point clouds is referred to as *registration*. The process involves in finding the relative positions and orientations of the separately acquired views in a global coordinate framework, such that the intersecting areas between them overlap perfectly.

Figure 4.1 – Biblical interpretation of Procrustes stretching his short guest to fit to the bed (picture taken from internet).

Registration is an essential component of 3D acquisition pipeline and is fundamental to computer vision, computer graphics and reverse engineering. Typically, the term *registration* is used for the geometric alignment of a pair or more 3D data point sets, while the term *fusion* is used when one wants to get a single surface representation from registered 3D datasets. The algorithms for these two problems are inherently different. Registration algorithms associate sets of data into a common coordinate system by minimizing the alignment error, however, the algorithms for *image registration* which have extensive applications in medical imaging are quite different. Throughout this dissertation, the word *registration* is liberally applied to *geometric registration* involving depth data.

The structure of the chapter is as follows: starting with a brief introduction and history of registration algorithms, the state-of-the-art methods are presented in the section after. ICP has been discussed elaborately as it has been extensively used in this thesis: for evaluating performance of the jump-edge filter as in previous chapter, for scan matching in SLAM and also in design of novel shape signature. In order to design a robust shape signature for object's depth images, a very stable keypoint detector is needed. There are dozens of keypoint detectors already available and more often it is confusing to chose among them as each of them have their own specific specialities and benefits. So, using ICP, the repeatability of different keypoint detectors are evaluated. The results of scan matching and repeatability are discussed in the last section of

this chapter and also in Chapters 5 & 6.

4.2 A Brief History of Registration Algorithms

Decades before the first registration algorithm was invented for scan matching, Hurley and Cattell (1962) presented registration as *Orthogonal Procrustes*[1] problem, although Orthogonal Procrustes analogy represents non-rigid diffeomorphic registration rather than rigid registration in typical scan matching. Faugeras and Hebert (1986) defined closed-form distances to minimize point-to-point and plane-to-plane alignment error, their method solved translation and rotation as two-step procedure. Later, Walker, Shao, and Volz (1991) resolved rotation and translation error using dual quaternions. During this time, Besl and McKay (1992) christened their registration algorithm as ICP (Iterative Closest Point), which soon going to be famous in robotics and computer vision and medical imaging community. They expressed the problem as:

> Given 3D data in a sensor coordinate system, which describes a data shape that may correspond to a model shape, and given a model shape in a model coordinate system in a different geometric shape representation, estimate the optimal rotation and translation that aligns, or registers, the model shape and the data shape minimizing the distance between the shapes and thereby allowing determination of the equivalence of the shapes via a mean-square distance metric.

However, the algorithm makes few assumptions which are not quite suitable for most of the applications involving registration. Firstly, each point in one scan has a corresponding point in the other one, which implies that the two scans to be registered are identical but are present spatially in different coordinate systems. Secondly, the proof of the solution's convergence is demonstrated under the assumption that the number of associated points, or their weight, remains constant. These problems where reported by Champleboux et al. (1992) while developing early registration solutions for medical applications. Chen and Medioni (1991) extended the point-to-point error metric of Besl and McKay (1992) to point-to-plane which is still quite used nowadays. Zhang (1993) pioneered the idea of using ICP-based solutions for outdoor robotic applications. He has highlighted few modifications to be made in ICP in order to be used for robotic applications. In the first half of 1990s and later, scientific community has seen legion of applications based on ICP: object reconstructions, non-contact inspections, medical and surgery support, organ and environment reconstructions, 3D modelling in digital cultural heritage and autonomous vehicle navigation, to name a few. ICP, due to its simplicity has become a popular technique, and researchers from past two decades have extended ICP to match their own idiosyncratic problems. As a result, there are close to 400 ICP related/variant articles in IEEE Xplore and are still growing every year. However, as there is no comparison framework for these variants of ICP, the selection of an appropriate version for particular experimental condition/problem is difficult. A complete history and detailed applied registration for robotics can be found in the thesis of Pomerleau (Pomerleau, 2013) and this section has been partially inspired from his dissertation.

4.3 State-of-the-Art Methods

Registration algorithms can be coarsely classified into rigid and non-rigid approaches. Rigid approaches assume a rigid environment such that the transformation (Euclidean transformation) can be modelled with only 6 Degrees of Freedom (DoF). Non-rigid methods deal non-rigid transformations (*similarity transform, affine transform* and *orthogonal projection*) and able to cope

[1] Trivia: Procrustes whose name means "he who stretches", was the most interesting of Theseus' challenges on the way to become a hero in Greek mythology. Procrustes was a host who adjusts his guests to their bed; either stretching the person if he is short or chopping of the limbs if the person is tall. In the end, Theseus made him have his own medicine, fitting Procrustes to his own bed (see Fig. 4.1).

with articulated objects or soft bodies that change shape over time. Non-rigid registration is more difficult than the rigid counterpart, as it not only faces the common problems of the latter but also need to deal with the deformation (e.g., morphing, articulation). Unlike the rigid case, where a few correspondences are sufficient to define one candidate rigid transformation for hypothesis testing, both deformation and alignment have to be answered in the non-rigid case without strong prior assumptions, often requiring a more reliable correspondences to be defined. Most state-of-the-art applications involving registration employ either a simple Singular Value Decomposition (SVD) or Principal Component Analysis (PCA) based registration, or use a more advance iterative scheme based on the Iterative Closest Point (ICP) algorithm. Recently, many variants on the original ICP approach have been proposed, the most important of which are non-linear ICP, generalized ICP, and non-rigid ICP.

4.3.1 Principal Component Analysis

PCA is widely used for classification and compression techniques to project data on new orthonormal basis in the direction of the largest variance (Yambor, Draper, and Beveridge, 2002). The direction of the largest variance corresponds to the largest eigenvector of the covariance matrix of the data, whereas the magnitude of this variance is determined by the corresponding eigenvalue. Therefore, if the covariance matrix of two point clouds differs from the identity matrix, a rough registration can be obtained by simply aligning the eigenvectors of their covariance matrices. For this, firstly, the two point clouds are centered such that the origins of their original bases coincide by subtracting the centroid coordinates from each of the point coordinates. In the second step the covariance matrix of each point cloud is calculated. And the final step involves the calculation of eigenvectors for both covariance matrices. The largest eigenvector is a vector in the direction of the largest variance of the 3D point cloud, and hence represents the point cloud's rotation. The problem of aligning two point clouds simplifies to aligning/rotating their eigenvectors with the largest magnitude.

4.3.2 Singular Value Decomposition (SVD)

In general, the scans obtained by the sensors for mapping are partially overlapping. PCA based registration simply aligns the directions of the largest variance of each point cloud and therefore does not minimize the Euclidean distance between corresponding points of the datasets. As a result, this technique is very sensitive to outliers and only works well if each point cloud is approximately normally distributed (Bellekens, Spruyt, and Maarten Weyn, 2014). However, if the point correspondences between the two scans are known, reducing the Euclidean distances between these set of pair of points simplifies to linear-least-square problem, which can be robustly solved by SVD (Marden S, 2012). A correlation matrix $\mathbf{M}$ is calculated for the two centred point clouds based on the correspondences. The eigenvalue decomposition is then given by:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}} \tag{4.1}$$

The optimal solution to the least-square problem is given by rotation and translation:

$$\mathbf{R}_{\mathbf{t}}^{\mathbf{s}} = \mathbf{U}\mathbf{V}^{\mathbf{T}} \tag{4.2}$$

$$\mathbf{t} = \mathbf{c}_{\mathbf{s}} - \mathbf{R}_{\mathbf{t}}^{\mathbf{s}}\mathbf{c}_{\mathbf{t}} \tag{4.3}$$

4.3.3 Iterative Closest Point

The Iterative Closest Point (ICP) though Iterative Corresponding Point is better abbreviation; has become the dominant method for aligning three-dimensional models and is purely based on the geometry and, sometimes, color of the meshes; and is the ***de facto*** standard for geometric alignment of three-dimensional models when an initial relative pose estimate is available. This

algorithm is widely used for registering the outputs of 3D scanners, which typically only scan an object from one direction at a time and has applications in architecture, industrial automation, agriculture, cultural heritage conversation, medical data processing, art history, archaeology, and search and rescue robotics. The ICP Algorithm was developed by Besl and McKay and presented in 1992 (Besl and McKay, 1992). It is developed to register two given sets of points or 3D shapes in a common coordinate system. The algorithm is an iterative two-step procedure, in each iteration, the algorithm selects the closest points as correspondences and calculates the transformation $(\mathbf{R},\mathbf{t})$, for minimizing the following equation:

$$E(R,t) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_d} w_{i,j}\left|\left|m_i - (Rd_j + t)\right|\right|^2 \tag{4.4}$$

where N_m and N_d are the number of points in the model set M and dataset D respectively and $w_{i,j}$ is assigned value of 1 if the i-th point of M describes the same point in space as the j-th point of value of D, otherwise $w_{i,j}$ is 0. (4.4) can be reduced to:

$$E(R,t) \quad \propto \quad \frac{1}{N}\sum_{i=1}^{N}||m_i - (Rd_i + t)||^2, \tag{4.5}$$

with $N = \sum_{i=1}^{N_m}\sum_{j=1}^{N_d} w_{i,j}$, since the correspondence matrix can be represented by a vector $\mathbf{v}$ containing the point pairs, *i.e.*, $\mathbf{v} = (\mathbf{p_1}, \mathbf{m}_{f(\mathbf{p_1})}), (\mathbf{p_2}, \mathbf{m}_{f(\mathbf{p_2})}), \ldots, (\mathbf{p}_{N_p}, \mathbf{m}_{f(\mathbf{p}_{N_p})})$, with $f(x)$ being the search function returning the closest point. The assumption is that, in the last iteration step, the point correspondences and therefore the vector of point pairs are correct (Nüchter, Lingemann, and Hertzberg, 2007). In each ICP iteration, the transformation can be calculated by one of these four methods which have similar performance and stability concerning noisy data (Lorusso, Eggert, and Fisher, 1995): a SVD based method of Arun, Huang, and Blostein (1987), a quaternion method of Horn (1987), an algorithm using orthonormal matrices of Horn, Hilden, and Negahdaripour (1988) and a calculation based on dual quaternions of Walker, Shao, and Volz (1991). The first step of the computation using SVD (Arun, Huang, and Blostein, 1987) method is to decouple the calculation of the rotation R from the translation t (the equations solved below are taken from the documentation of Borrmann et al. (2008a) with permission). This can be done using the centroids of the points belonging to the matching:

$$c_m = \frac{1}{N}\sum_{i=1}^{N} m_i, \qquad\qquad c_d = \frac{1}{N}\sum_{i=1}^{N} d_j \tag{4.6}$$

and

$$M' \quad = \quad \{m'_i = m_i - c_m\}_{1,\ldots,N}, \tag{4.7}$$

$$D' \quad = \quad \{d'_i = d_i - c_d\}_{1,\ldots,N}. \tag{4.8}$$

After replacing (4.6), (4.7) and (4.8) in the error function, $E(R,t)$, (4.5) becomes:

$$\begin{aligned} E(R,t) \propto \frac{1}{N}\sum_{i=1}^{N}||m'_i - Rd'_i - \underbrace{(t - c_m + Rc_d)}_{=\tilde{t}}||^2 \\ = \frac{1}{N}\sum_{i=1}^{N}\left|\left|m'_i - Rd'_i\right|\right|^2 \qquad (4.9a) \\ -\frac{2}{N}\tilde{t}\cdot\sum_{i=1}^{N}\left(m'_i - Rd'_i\right) \qquad (4.9b) \\ +\frac{1}{N}\sum_{i=1}^{N}||\tilde{t}||^2. \qquad (4.9c) \end{aligned}$$

In order to minimize the sum above, all terms have to be minimized. The second sum (4.9b) is zero, since all values refer to centroid. The third part (4.9c) has its minimum for $\tilde{t} = \mathbf{0}$ or

$$t = c_m - Rc_d. \tag{4.10}$$

Therefore, the algorithm has to minimize only the first term, and the error function is expressed in terms of the rotation only:

$$E(R,t) \propto \sum_{i=1}^{N} \left|\left| m'_i - Rd'_i \right|\right|^2. \tag{4.11}$$

The rotation ($R = VU^T$) can be calculated from the SVD of covariance matrix H ($H = U\Lambda V^T$):

$$H = \sum_{i=1}^{N} m_i'^T d'_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}, \tag{4.12}$$

with $S_{xx} = \sum_{i=1}^{N} m'_{ix} d'_{ix}$, $S_{xy} = \sum_{i=1}^{N} m'_{ix} d'_{iy}$, ...

SVD based ICP has been used throughout this thesis either through PCL ICP (Rusu and Cousins, 2011) or 3DTK (Borrmann et al., 2008a).

Since the introduction of ICP by Chen and Medioni (1991) and Besl and McKay (1992), many variants have been introduced on the basic ICP concept. These variants have evolved from one of six stages of the algorithm (Rusinkiewicz and Levoy, 2001).

1. **Selection** of some set of points in or both meshes.
2. **Matching** these points to samples in the other mesh.
3. **Weighting** the corresponding pairs appropriately.
4. **Rejecting** certain pairs based on looking at each pair individually or considering the entire set of pairs.
5. **Error-Metric** assignment on the point pairs.
6. **Minimizing** the error metric.

Rusinkiewicz and Levoy propose a high speed ICP variant using a point-to-plane error metric (Neugebauer, 1997) and a projection-based method to generate point correspondences (Blais and Levine, 1995). And they also conclude that the other stages of the ICP process appear to have little effect on the convergence rate (Nüchter, Lingemann, and Hertzberg, 2007) and most state-of-the-art methods are based on the last two stages of the ICP. Two widely used ICP variants are the ICP point-to-point and the ICP point-to-surface algorithms. These approaches only differ in their definition of point correspondences.

4.3.3.1 ICP Point-to-Point

An important step in ICP is finding the point correspondences between the two scans. ICP point-to-point simply obtains point correspondences by searching for nearest neighbour in target d_i to the source point cloud point m_i. Throughout this document, *source point cloud*, *Model set* and *reference cloud* have been used to describe same concept/object; and *target point cloud*, *Data set*, *data cloud* are their synonymous counterparts. The nearest neighbour matching is defined in terms of the Euclidean distance metric:

$$\hat{\imath i} = \arg\min_{i} \|m_i - d_i\|^2 \tag{4.13}$$

where $i \in [0, 1 \ldots N]$, and N represent the number of points in the target point cloud. Similar to (4.3.2) approach, the rotation R and translation t are estimated by minimizing the squared distance between these corresponding pairs:

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_{i=1}^{N} \|(Rm_i + t) - d_i\|^2 \tag{4.14}$$

ICP, then iteratively solves (4.13) and (4.14) until the error becomes smaller than a threshold or it stops changing. Besl and McKay demonstrated that the iteration terminates in a minimum (Besl and McKay, 1992), however, generally, implementation of ICP would use a maximal distance for closest points to handle partially overlapping point sets. In this case, the proof in (Besl and McKay, 1992) does no longer hold, since the number of points as well as the value of $E(R,t)$ might increase after applying a transformation (Nüchter, Lingemann, and Hertzberg, 2007).

4.3.3.2 ICP Point-to-Surface

The point correspondences procedure of ICP *point-to-point* is very sensitive to outliers. Chen and Medioni (1991) used a point-to-plane error metric in which the object of minimization is the sum of the squared distance between a point and the tangent plane at its correspondence point. Unlike the point-to-point metric, which has a closed-form solution, the point-to-plane metric is usually solved using standard non-linear least squares methods, such as the Levenberg-Marquardt method. This method assumes that the point clouds are locally linear, such that the local neighbourhood of a point is co-planar. This local surface can then be defined by its normal vector $\mathbf{n}$, which is the smallest eigenvector of the covariance matrix of neighbourhood d_i. Then object of the minimization is the sum of the squared distance between each source point and the tangent plane at its corresponding target point. More specifically if $m_i = (m_{ix}, m_{iy}, m_{iz}, 1)^T$ is a source point, $d_i = (d_{ix}, d_{iy}, d_{iz}, 1)$ is the corresponding target point, and $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz})$ is the unit normal vector at d_i, then R and t can be calculated by:

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_{i=1}^{N} \left(\|((Rm_i + t) - d_i) \cdot \mathbf{n}_i\| \right)^2 \tag{4.15}$$

This method is also been referred as *normal shooting* (Rusinkiewicz and Levoy, 2001) and can also be linearized by assuming small incremental rotations, *i.e.*, $\sin\theta \approx \theta and \cos\theta \approx \theta$.

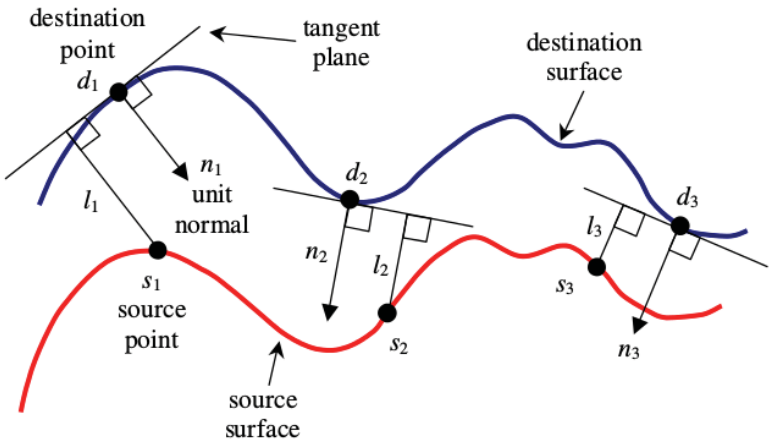


Figure 4.2 – Point-to-plane error metric. Source: (Low, 2004).

4.3.3.3 Generalised ICP (GICP)

Segal, Haehnel, and Thrun (2009) have combined the ICP and ICP *point-to-point* algorithms into a single probabilistic framework. This framework is used to model locally planar surface structure from both scans instead of just the source scan as it is done in point-to-plane. This method is also called as *ICP plane-to-plane*. Traditional ICP assumes that the source point cloud is taken from a known geometric surface instead of being obtained through noisy sensor measurements. Due to discretization errors it is not possible to obtain point-to-point matching. Point-to-plane, as it allows point offsets along the surface on the target point cloud, has reduced discretization error to some extent. However, in GICP source point cloud $A = \{\mathbf{a}\}_i$ and the target point cloud $B = \{\mathbf{b}\}_i$ are assumed to consist of random samples from an underlying unknown point cloud $\hat{A} = \{\hat{\mathbf{a}_i}\}$ and $\hat{B} = \{\hat{\mathbf{b}_i}\}$. For the underlying and unknown point clouds $\hat{A}$ and $\hat{B}$, perfect correspondences exist, whereas this is not the case for the observed point clouds A and B, since each point $\mathbf{a_i}$ and $\mathbf{b_i}$ is assumed to be sampled from normal distribution such that $\mathbf{a_i} \sim \mathcal{N}(\hat{\mathbf{a}_i}, C_i^A)$ and $\mathbf{b_i} \sim \mathcal{N}(\hat{\mathbf{b}_i}, C_i^B)$. The covariance matrices C_i^A and C_i^B are unknown. If both point clouds would consist of deterministic samples from known geometric models, then both covariance matrices would be zero such that then $A = \hat{A}$ and $B = \hat{B}$. In the following, let T be the affine transformation matrix that maps from $\hat{A}$ to $\hat{B}$ such that $\hat{\mathbf{b}_i} = \hat{T}_i \hat{\mathbf{a}_i}$. If T would be known, we could apply this transformation on the observed point cloud A, and define the error to be minimized as $d_i^T = \mathbf{b_i} - T_i \mathbf{a_i}$. d_i^T is also drawn from normal distribution, as it is a linear combination of $\mathbf{a_i}$ and $\mathbf{b_i}$ which are assumed to be drawn from independent normal distribution:

$$d_i^T \sim \mathcal{N}(\hat{\mathbf{b}_i} - T_i \hat{\mathbf{a}_i}, C_i^B + T C_i^A T^{\mathbf{T}}) \tag{4.16}$$

$$= \mathcal{N}(0, C_i^B + T C_i^A T^T) \tag{4.17}$$

The optimal transformation $\hat{N}$ is then the transformation that minimizes the negative log-likelihood of the observed errors d_i:

$$\hat{T} = \arg\min_T \sum_i \log\left(p\left(d_i^T\right)\right) = \arg\min_T \sum_i d_i^{T\mathbf{T}} \left(C_i^B + T C_i^A T^T\right)^{-1} d_i^T \tag{4.18}$$

Segal, Haehnel, and Thrun showed that both point-to-point and point-to-plane are specific cases of (Segal, Haehnel, and Thrun, 2009), only varying in the choice of covariance matrices C_i^A and C_i^B; if the source point cloud is assumed to be obtained from known geometric surface, $C_i^A = 0$. Furthermore, if the points in the target point cloud are allowed three degrees of freedom, then $C_i^B = I$. In this case, (4.18) reduces to:

$$\hat{T} = \arg\min_T \sum_i d_i^{T\mathbf{T}} d_i^T = \arg\min_T \sum_i \| d_i^T \|^2 \tag{4.19}$$

which is exactly the optimization problem that is solved by traditional point-to-point ICP algorithm. Similarly, C_i^A and C_i^B can be chosen such that obtaining the maximum likelihood estimator corresponds to minimizing the point-to-plane or the plane-to-plane distances between both point clouds (Bellekens, Spruyt, and Maarten Weyn, 2014).

4.4 Demonstration of Examples

ICP point-to-point from PCL has been used to determine robust 3D keypoint detectors for designing a shape signature based on keypoint distributions. Two 3D views of an object are brought to the same coordinate system using ICP (see Fig. 4.3) and then the number of keypoints which are repeated on both views are counted using nearest neighbours method.

3D scan matching has been used twice in the algorithm to obtain globally consistent mapping. Once for initial registration of all the sequentially obtained scans and then later to optimize the graph of scans in GraphSLAM. Figure 4.4 shows the registration of three sequentially obtained

scans using ICP from (Borrmann et al., 2008a). ICP fitness score as used in previous chapter to evaluate the quality of jump edge filter, is also used from PCL.

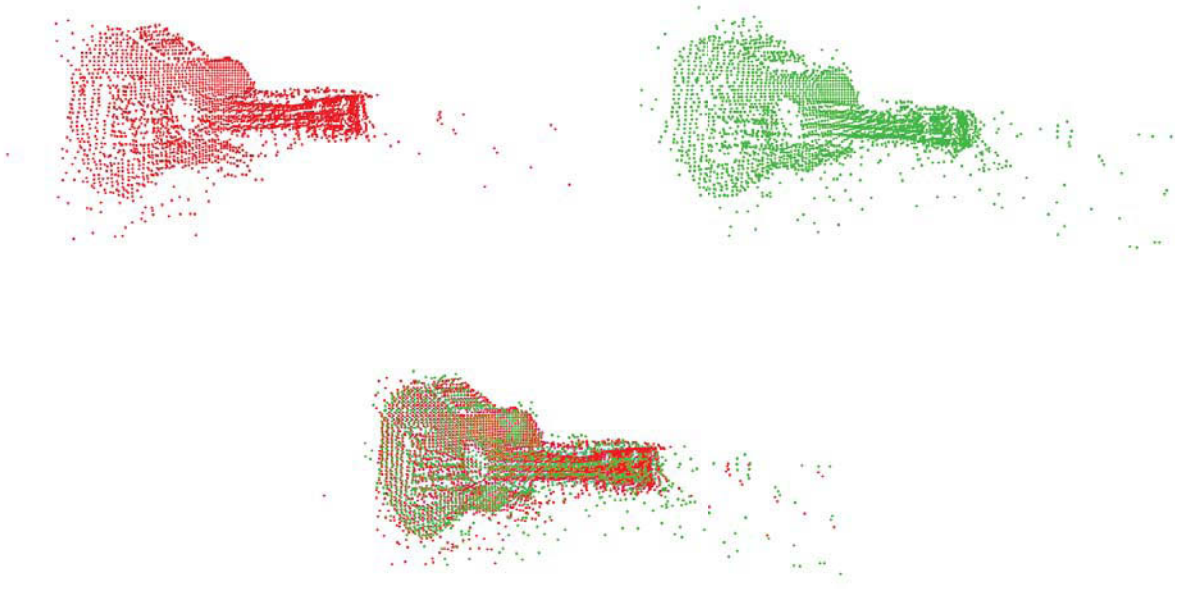


Figure 4.3 – Two different views of a Guitar, one rotated $5.000^{\circ\prime\prime\prime}$ from the other, are aligned in same coordinate system using ICP. And using the nearest neighbour method with one view as source, the keypoints which are repeated are counted to determine repeatability of keypoint detectors.

4.5 Conclusion

In this chapter, different scan matching approaches have been discussed. ICP is the most popular and widely used method for registration of 2D/3D surfaces, so ICP and its variants are discussed elaborately. Singular Value Decomposition (SVD) based ICP has been used in this thesis at multiple stages. First, to evaluate the performance of novel jump edge filter for scan registration using ICP-fitness score. Second, to calculate the repeatability of different 3D keypoint detectors. Third and fourth for scan matching of sequentially acquired scans and graph optimization of poses respectively. The next chapter discusses the last two stages where ICP is used for SLAM problem. The sequentially acquired scans are brought into one global coordinate system by registration of n^{th} and $(n+1)^{th}$ scans in the sequence. And the map is optimized by a threshold on the pose error using ICP again.

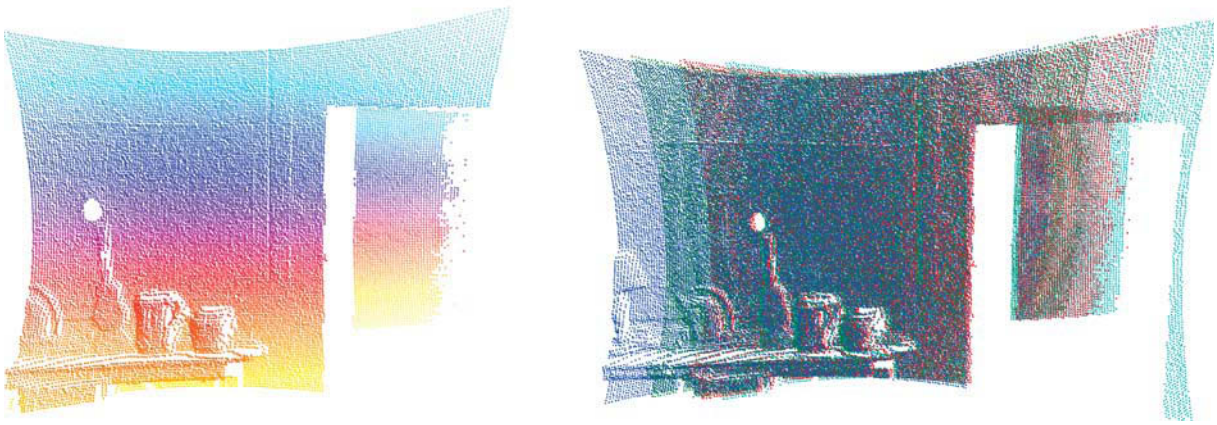


Figure 4.4 – 3D scan matching for mapping indoor environment. **Left** The middle scan of the three consecutive scans taken of a scene having several objects on a table. **Right** After scan matching the three consecutive views are aligned in the same coordinate system. Note that the objects in the registered scan get increasingly noisy as the number of scans are increased and hence it is difficult to semantic label them in the global map.

Chapter

5

Globally Consistent Mapping using GraphSLAM

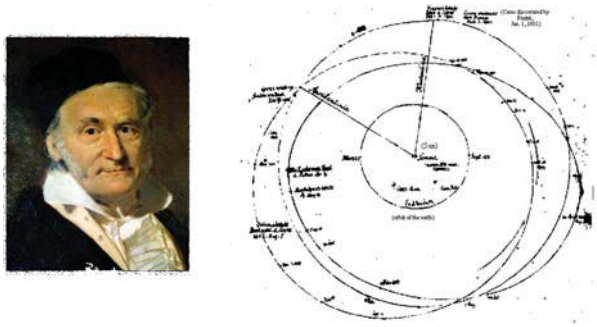
This chapter starts with a brief history of SLAM problem and evolution of solutions to it. The state-of-the-art techniques in SLAM are presented in and out. A simple unknown indoor environment with relatively unfluctuating lighting conditions is considered for mapping and localization. The camera positioned on a mobile tripod, is ready to capture images at prearranged locations in the environment. The prearranged locations are in fact used as ground truth for estimating the variance with calculated poses from SLAM, and also as initial pose estimates for ICP. Interesting point is that, in this thesis, any type of Inertial Measurement Units or visual odometry techniques or explicit loop closures have not been utilized, given the fact that, data from time-of-flight camera is extremely noisy and sensitive to external conditions (such as lighting, transparent surfaces, parallel overlapping surfaces etc.). The whole SLAM dataset acquired with 5 m and 10 m is publicly available for academic research.

5.1 Introduction and Background

The genesis of SLAM problem can be tracked back to a discussion at 1986 IEEE Robotics and Automation Conference held in San Francisco. Over the course of the conference, many paper table cloths and napkins were filled with long discussion about consistent mapping. The result of this conversation was a recognition that consistent probabilistic mapping was a fundamental problem in robotics with major conceptual and computational issues that needed to be addressed (Durrant-Whyte and Bailey, 2006). The acronym "SLAM" and its structure was first coined in a mobile robotics survey paper presented at 1995 International Symposium on Robotics Research (Durrant-Whyte, Rye, and Nebot, 1996):

> *The Simultaneous Localization and Mapping (SLAM) problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map* (Durrant-Whyte and Bailey, 2006).

However, technically speaking, the historical roots of SLAM can be dated back to 1809; Gauss in (Gauss, 1877; Gauss and Davis, 2004) invented least-squares method to explain and calculate the elliptical trajectories (conics precisely) of planets orbiting the sun (see Fig. 5.1)[1]. Which in robotic jargon is, mapping of planets position in temporal and spatial space around the sun.



.

Figure 5.1 – Carl Friedrich Gauss, German mathematician known for calculating the planetary motion around the sun. He successfully calculated the trajectory of Ceres and Pallas (sketch by him, courtesy of Universitätsbibliothek Göttingen) using *least-squares method* and universal gravitation concept by Sir Isaac Newton (Newton et al., 1687) and Kepler's laws of planetary motion (Kepler, 1609)

Suppose, we want to estimate an unknown variable χ (it includes the trajectory of the robot as discrete set of poses (X) and the position of the landmarks in the environment, see Fig. 5.2), given a set of measurements $Z = \{z_k : k = 1, \ldots, m\}$, such that each measurement can be expressed as function of χ, *i.e.*, $z_k = h_k(\chi_k) + \varepsilon_k$, where $h_k(\cdot)$ is the measurement or observation model, ε_k is random measurement noise and $\chi_{1:k}$ is transition state model ($\chi_k \subset \chi$). This can be effectively solved using Maximum a Priori (MAP) estimation. In MAP estimation, χ is estimated by computing the assignment of variables χ^* that attains the maximum of the posterior $p(\chi|Z)$:

$$\chi^* \doteq \arg\max_{\chi} p(\chi|Z) = \arg\max_{\chi} p(Z|\chi)\, p(\chi) \tag{5.1}$$

where, $p(Z|\chi)$ is the likelihood of the measurements Z, given the assignment χ, and $\mathbf{p}(\chi)$ is a prior probability over χ. The $p(\chi)$ includes any prior knowledge about χ; if there is no prior information available, then $p(\chi)$ becomes uniform distribution and MAP estimation reduces to *maximum likelihood estimation*.

[1] The complete description of the Gauss' method to calculate Ceres orbit is given in ("How Gauss determined the orbit of Ceres," 1998)

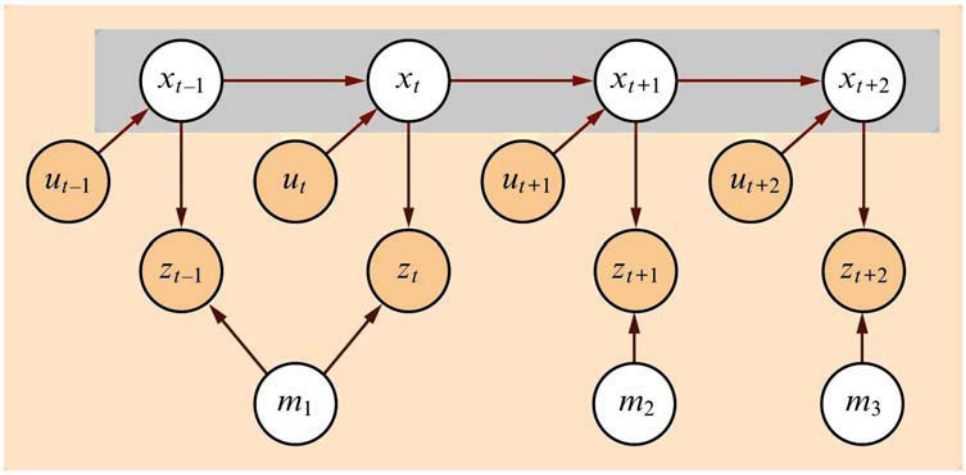


Figure 5.2 – The SLAM problem depicted as a Bayes Network Graph. The robot moves from location (x_{t-1}) to location (x_{t+2}), driven by a sequence of controls. At each location (x_t), it observes a nearby feature in the map $m = \{m_1, m_2, m_3\}$. $U_T = \{u_1, u_2, u_3, \ldots, u_T\}$ are odometry readings and $Z_T = \{z_1, z_2, z_3, \ldots, z_T\}$ are measurement readings. Source: (Thrun and Leonard, 2008).

Assuming that the measurements Z are independent and identically distributed, (5.1) reduces to:

$$\chi^* = \arg\max_{\chi} \mathbf{p}(\chi) \prod_{k=1}^{m} p(z_k|\chi) = \arg\max_{\chi} p(\chi) \prod_{k=1}^{m} p(z_k|\chi_k) \tag{5.2}$$

Making another assumption, that measurement noise, ε_k, is a zero-mean Gaussian noise with the information matrix Ω_k (inverse of covariance matrix), then the measurement likelihood becomes:

$$p(z_k|\chi_k) \propto \exp(-\frac{1}{2} e^T \Omega e), \tag{5.3}$$

where,

$$e = h_k(\chi_k) - z_k \tag{5.4}$$

As maximizing the posterior is the same as minimizing the negative log-posterior

$$\chi^* = \arg\min_{\chi} -\log\left(p(\chi) \prod_{k=1}^{m} p(z_k|\chi_k)\right) = \arg\min_{\chi} \sum_{k=0}^{m} ||h_k(\chi_k) - z_k||^2_{\Omega_k}, \tag{5.5}$$

which is a standard minimization problem that can be solved by non-linear least squares method. Gauss applied the same method to track the orbit of dwarf planet Ceres, which is equivalent to estimating the robot trajectory from measurement data. SLAM is most often formulated as MAP estimation problem. Gauss' minimization technique has been applied to a number of problems in all branches of sciences; including surveying (Golub and Plemmons, 1980), photogrammetry (Brown, 1976; Granshaw, 1980; Slama, Theurer, and Henriksen, 1980; Cooper and Robson, 1996) and computer vision (Faugeras, 1993; Szeliski and Kang, 1994; Triggs et al., 2000; Hartley and Zisserman, 2004). However, it is popularised with *bundle adjustment* (in photogrammetry) and *structure from motion* (in computer vision) terms.

According to Cadena et al. (2016), SLAM research history can be bifurcated into *classical age* (1986 – 2004) and *algorithmic-analysis age* (2004 – 2015). The classical age saw the introduction of the main probabilistic formulations for SLAM, including approaches based on Kalman Filters (Smith, Self, and Cheeseman, 1990; Castellanos et al., 1999), Rao-Blackwellised Particle Filters (Hähnel et al., 2003; Grisetti, Stachniss, and Burgard, 2007; Montemerlo et al., 2002b), Information

Filters (Thrun et al., 2004; Eustice, Singh, and Leonard, 2006) and maximum likelihood estimation. The first three SLAM formulations are excellently described in (Thrun, Burgard, and Fox, 2005) and (Thrun and Leonard, 2008). They are also called as *filtering*[2] methods or on-line SLAM systems; model the problem as an on-line state estimation, where the state of the system consists in the *current* robot position and the map. The estimate is modulated or refined by the on-coming new measurements and hence recursive, incremental.

5.1.1 On-Line State Estimation

On-line state estimation, also called as recursive state estimation, address the problem of estimating quantities from sensor data that are not directly observable, but can be inferred. In most robotic applications, determining what to do is relatively simple if one only knew certain parameters, like robot navigation is very easy if it knows where it is presently and where are the nearby obstacles/landmarks. However, these parameters are not measurable, and robot has to depend on its sensors and recursively estimate its state as the sensor measurements are corrupted by noise. It should be taken into account that, previous notations in this chapter for states, odometry and measurements is not continued here-after.

5.1.1.1 Gaussian Filters

Gaussian filters are family of filters that recursively estimate the state (x) when the beliefs (or posteriors) are represented by multivariate normal distributions:

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{5.6}$$

They are the most popular family of techniques despite their several shortcomings. Also called as parametric filters as the density over the variable x is characterized by mean (μ) and a symmetric and positive semi-definite covariance matrix (Σ). Kalman Filters (*KF*) is one of the best studied Gaussian filter, introduced in the 1950s by Rudolph Emil Kálmán (Kalman, 1960). It computes belief, $bel(x_t)$ at time t for continuous states using first and second moments representation (μ, Σ, u). The input for *KF* is $bel(x_{t-1})$ with μ_{t-1} and Σ_{t-1} parameters. It updates these parameters using the control u_t and the measurement z_t and outputs $bel(x_t)$.

Algorithm 2 Kalman Filter Algorithm

1. Input: $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:
2. $\bar{\mu}_t = A_t\mu_{t-1} + B_t u_t$
3. $\Sigma_t = A_t\bar{\Sigma}_{t-1}A_t^T + R_t$
4. $K_t = \bar{\Sigma}_t C_t^T \left(C_t\bar{\Sigma}_t C_t^T + Q_t\right)^{-1}$
5. $\mu_t = \bar{\mu}_t + K_t\left(z_t - C_t\bar{\mu}_t\right)$
6. $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
7. Output: μ_t, Σ_t

Kalman filter is not applicable for non-linear state transitions (with non Gaussian noise), in this case EKF (Extended Kalman Filter) comes to rescue. EKF assumes that the next state

[2]Filter is just a fancy word for an algorithm that takes an input (typically, a sensor signal) and calculates a function of that input.

probability and measurement probabilities are governed by non-linear functions g and h instead of linear A_t and B_t:

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{5.7}$$

Kalman Filter

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t, \tag{5.8}$$
$$z_t = h(x_t) + \delta_t \tag{5.9}$$

Extended Kalman Filter

However, sometimes non-linear systems are linearized at the expense of diminished estimation performance. The linearization errors can be mitigated by reducing the degree of non-linearity by augmenting the actual non-linear measurement model with additional, properly chosen mappings as proposed in (Liu and Li, 2013). One such method which mitigates these errors is *statistical linear regression* (again Gauss' contribution!), and the non-linear Kalman Filters which use this technique are called as Linear Regression Kalman Filters (LRKF) (Lefebvre, Bruyninckx, and Schutter, 2004; Lefebvre, Bruyninckx, and Schutter, 2005; Steinbring and Hanebeck, 2015; Ulas and Temeltas, 2014). On the other hand, EKF accommodates the non-linearities from the real world, by approximating the robot motion model using linear functions (Davison and Murray, 2002; Leonard and Newman, 2003; Jensfelt et al., 2006; Se, Lowe, and Little, 2002). Approximation of Gaussian plays an important role for non-linear system to improve state estimation. EKF can be viewed as a first-order approximation to the optimal solution; in which the state distribution is approximated by Gaussian Random Variable (GRV) at the cost of introducing large errors in the true posterior mean and covariance. The Unscented[3] Kalman Filter (UKF) (Julier and Uhlmann, 2004) is another filtering technique which uses unscented transform rather than linearization; but applies GRV to represent the state distributions by carefully choosing a minimal set of carefully chosen sample points. The Cubature rule for approximation of Gaussian (CKF Cubature Kalman Filter) proposed by (Arasaratnam and Haykin, 2009) provides more accurate results and solves large spectrum of non-linear problems. Pakki (2013) proposed CKF-SLAM using point features.

The dual of Kalman Filter is Information Filter (IF), which also represents the belief as Gaussian like KF and EKF; the only difference is the way it is represented. IF represents Gaussian beliefs as information matrix (inverse of covariance matrix, Ω) and an information vector (ξ).

$$\Omega = \Sigma^{-1} \tag{5.10}$$
$$\xi = \Sigma^{-1}\mu \tag{5.11}$$

Up on some straightforward sequence of transformation on (5.6) leads to a belief:

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \frac{1}{2}\mu^T\Sigma^{-1}\mu\right\} \tag{5.12}$$
$$= \underbrace{\det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mu^T\Sigma^{-1}\mu\right\}}_{\text{const.}} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\} \tag{5.13}$$
$$= \eta \exp\left\{-\frac{1}{2}x^T\Omega x + x^T\xi\right\} \tag{5.14}$$

[3] Trivia: The use of the word "unscented" rather than Uhlmann filter has quite a bit history; Uhlmann was working in the lab alone while everyone of his colleagues were partying in Royal Opera House. He happens to notice someones' deodorant on a desk and came up with the term Unscented Kalman Filter

After negative log and setting the derivative of the belief to zero:

$$x = \Omega^{-1}\xi, \tag{5.15}$$

since Ω is symmetric positive semi-definite and -log$(p(x))$ is quadratic distance function with mean $\mu = \Omega^{-1}\xi$.

The information matrix (Ω) determines the rate at which the distance function increases in the different dimensions of the state (x).

Most of the filter based SLAM approaches use the image features or extract landmarks (Nieto, Bailey, and Nebot, 2007; Thrun, Burgard, and Fox, 1998; Castellanos and Tardos, 2000; Thrun et al., 2004) or convert 3D point clouds to 2D depth images (Li and Olson, 2011) or extract planes and use them for mapping (Zhang, Chen, and Liu, 2016; Pathak et al., 2010).

5.1.1.2 Non-Parametric Filters

Non-parametric filters do not rely on fixed functional form of posteriors, instead, they approximate posteriors by a finite number of values, each roughly corresponding to a region in the state space. Histogram filters decompose the state space into finitely many regions and the cumulative posterior is represented with single probability value:

$$range(X_t) = x_{1,t} \cup x_{2,t} \cup x_{3,t} \cup \ldots x_{K,t} \tag{5.16}$$

$$p(z_t|x_{k,t}) = \frac{p(z_t, x_{k,t})}{p(x_{k,t})} \tag{5.17}$$

Particle filters (also called the Sequential Monte-Carlo, SMC method) on other hand like histogram filters approximate posterior by finite number of parameters; but it differs in the way these parameters are generated and how they populate the state space. The particle filter (PF) represents the belief $bel(x_t)$ by a set of random state samples drawn from this posterior, which makes it handle high non-linear systems and non-Gaussian noise. Instead of representing the distribution by a parametric form (the exponential function that defines the density of a normal distribution), particle filters represent a distribution by a set of samples drawn from this distribution; the samples of a posterior distribution are called *particles* ($x_t^{[m]}$):

$$X_t := x_t^{[1]}, x_t^{[2]}, x_t^{[3]}, \ldots, x_t^{[M]} \tag{5.18}$$

This representation however has severe computational complexity on the state dimension, hence not suitable for real-time applications and map-building; but only for localization. There are few methods which combine PF with other strategies for the whole SLAM framework (Montemerlo et al., 2003; Montemerlo et al., 2002b; Hähnel et al., 2003; Blanco, Fernandez-Madrigal, and Gonzalez, 2007; Havangi, 2017; Xu et al., 2017), called as FastSLAM methods. FastSLAM takes advantage of an important property of SLAM problem: landmark estimates are conditionally dependent given robot's path (Montemerlo and Thrun, 2007), it decomposes the SLAM problem into a robot localization problem and set of landmark estimation problems that are conditioned on robot pose estimate. In FastSLAM, each particle makes its own local data association and PF is applied to sample over robot paths, eventually leading to less memory usage and faster computation.

5.1.1.3 Maximum Likelihood Methods

Maximum likelihood methods, also called as Expectation Maximization (EM) method, is an iterative statistical algorithm offers optimal solution with an expectation step (E-step) and maximization step (M-step). In E-step the posterior over robot poses is calculated for a given map

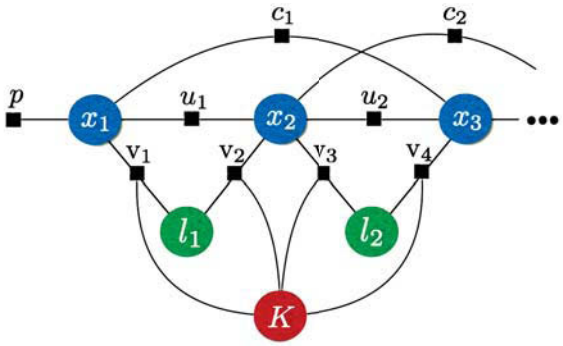


Figure 5.3 – Formulation of Full SLAM as factor graph. Blue circles denote the robot poses at consecutive time steps $(x_1, x_2, x_3, \dots)$, landmark positions are indicated with green circles $(l_1, l_2, l_3, \dots)$, red circles represent the variable associated with intrinsic calibration parameters (K). Factors are shown as black squares: the label u marks the factors corresponding to odometry readings, v marks factors corresponding to camera observations. c & p denotes loop closures and priors respectively. Source: (Cadena et al., 2016).

and in M-step, a map is calculated based on these pose expectations. The result of these two iterative steps is fine and accurate maps over the time and also the data association problem is very well handled. However, It should be noted that this method is ideal for map-building not for localization (Burgard et al., 1999) and moreover it is not suitable for real-time applications as it lacks incremental nature (Chen, Samarabandu, and Rodrigo, 2007). For these reasons, EM is usually combined with PF: use EM to construct map (M-step) and perform localization using different means like PF-based localizer to estimate poses from odometer readings (Thrun, 2002).

5.1.2 Modern SLAM Systems

On the other hand *algorithmic period* saw the study of fundamental properties of SLAM, including observability, convergence, consistency and sparsity. Most of the modern SLAM systems fall under the category of *smoothing approaches* (Chatila and Laumond, 1985; Lu and Milios, 1997a; Lu and Milios, 1997b; Gutmann and Konolige, 2000; Konolige, 2004; Eustice, Singh, and Leonard, 2006; Dellaert and Kaess, 2006; Folkesson and Christensen, 2004; Grisetti et al., 2007; Kaess et al., 2012; Kaess, Ranganathan, and Dellaert, 2008; Olson, Leonard, and Teller, 2006; Thrun and Montemerlo, 2006; Deans and Hebert, 2001; Duckett, Marsland, and Shapiro, 2002; Howard, Mataric, and Sukhatme, 2001; Frese and Duckett, 2003; Folkesson and Christensen, 2004; Folkesson, Jensfelt, and Christensen, 2005; Frese, 2004; Frese, 2006). Some of these only optimize/smooth the robot's trajectory, while others called as "full SLAM systems" (also popularly referred as *GraphSLAM, factor graph optimization, full smoothing, pose graph optimization* (see Fig. 5.3)), try to optimally estimate the entire set of sensor poses along with the parameters of all the features in the environment. The latter also called as SAM (simultaneous Smoothing And Mapping) (Thrun, Burgard, and Fox, 2005) rely on least-square error minimization (smoothing) technique. All these approaches project the SLAM problem as Maximum a Priori estimation (MAP) problem. The factors in (5.5) are not constrained to model projective geometry like in *Bundle Adjustment*, but includes a variety of sensor models. For instance, in laser-based mapping, the factors usually constrain relative poses corresponding to different viewpoints. Successive linearization methods (Gauss-Newton, Levenberg-Marquardt) are the typical methods to solve (5.5). Starting from an initial guess $\hat{\chi}$, approximate the cost function at $\hat{\chi}$ with a quadratic cost, which can be optimized in close form by solving a set of *normal equations*. In the modern SLAM systems, the matrices in normal equations are sparse, and their sparsity is dictated by the topology of their factor graph. There are many SLAM libraries which can solve tens of thousands of variables in fraction of minutes (Dellaert, 2012; Kümmerle et al., 2011; *Ceres Solver*; Kaess, Ranganathan, and Dellaert, 2008;

Incremental Block Cholesky Factorization for Nonlinear Least Squares in Robotics). The MAP based SLAM solvers are more accurate and efficient than original approaches based on non-linear filtering. However, some SLAM systems based on EKF (Extended Kalman Filters) (Mourikis and Roumeliotis, 2007; Hesch et al., 2014; Kottas et al., 2013) have demonstrated state-of-the-art performance.

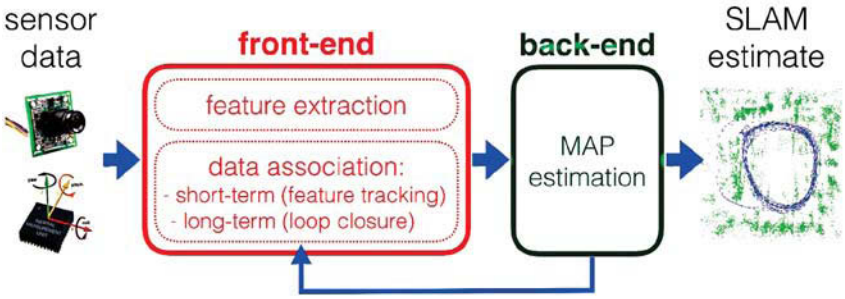


Figure 5.4 – Front-end and back-end in a typical modern SLAM system. Source: (Cadena et al., 2016).

The anatomy of modern SLAM system of algorithmic age has two main components: the *front-end* and the *back-end* (see Fig. 5.4). The front end abstracts sensor data into models that are amenable for estimation, while the back-end performs inference on the abstracted data produced from the front-end. Modern SLAM systems extensively depend on sensor data for feature extraction, accurate representation of the environment and for semantic interpretation of the scene. However, the output of the back-end, a map, can be parametrized as a set of spatially located landmarks, by dense representations like occupancy grids, surface maps or by raw sensor measurements. The choice of a particular map representation depends on the sensor used, on the characteristics of the environment, and on the estimation algorithm. Dense map representations like surface maps, point cloud and occupancy grids use range sensors (see Fig. 5.5). This thesis uses dense map representation using colorless point clouds obtained by time-of-flight camera.

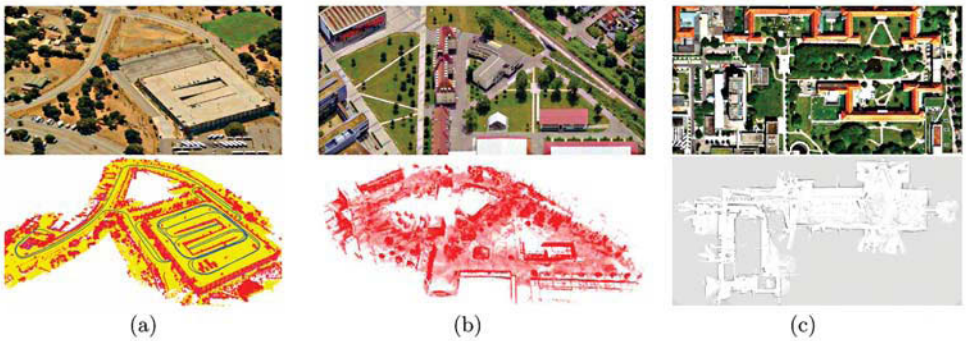


Figure 5.5 – (**a**) (bottom) A 3D map of the Stanford parking garage (top) aerial view. (**b**) Point cloud map acquired at the university of Freiburg. (**c**) Occupancy grid map acquired at the hospital of Freiburg. Grey: unobserved regions, white: traversable space, black: occupied regions. Source: (Grisetti et al., 2007).

5.1.3 Age of Robust-perception

Cadena (Cadena et al., 2016) speculated that SLAM is entering into a third era, the *robust-perception age*, which is characterized by the following key requirements:

- ▷ **Robust performance**: the SLAM system operates with low failure rate for an extended period of time in a broad set of environments; equipped with fail-safe mechanisms and self-tuning capabilities;
- ▷ **High-level understanding**: the SLAM system goes forwards to obtain high-level understanding of the environment (e.g., high-level geometry, semantics, physics);
- ▷ **Resource awareness**: the SLAM system is tailored to the available sensing and computational resources, and provide means to adjust the computational load depending on the available resources;
- ▷ **Task-driven perception** the SLAM system is able to select relevant perceptual information and filter out irrelevant sensor data, in order to support the task the robot has to perform; moreover, the SLAM system produces adaptive map representations, whose complexity may vary depending on the task at hand.

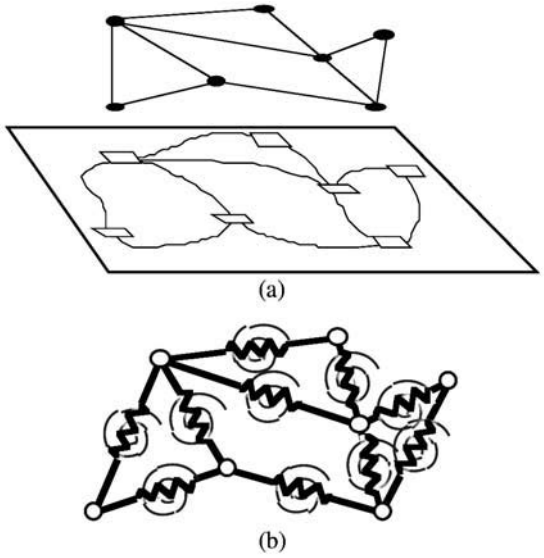


Figure 5.6 – (**a**) Graph-based representation of the environment (**b**) Equivalent spring model. Source: (Golfarelli, Maio, and Rizzi, 1998).

In this thesis, similar to modern SLAM system (see Fig. 5.4) except feature tracking instead scan matching is followed. This chapter presents an established globally consistent scan matching using GraphSLAM, starting with a brief history of SLAM origin and different types of methodologies are discussed in Section 5.1. And then the main idea of graph based SLAM is presented and complete mathematical derivation of it is detailed. The indoor environment which has been considered for semantic mapping is displayed and data acquisition process is also discussed in the same section. In Section 5.4 the results of the SLAM is given and a novel evaluation metric is then suggested. The SLAM output is a single reconstructed point cloud of the environment with estimated camera poses overlaid on it. The estimated poses are compared with the ground truth using a novel context-based similarity score metric. This section and partial amount of next section has been inspired by Cadena et al. (2016), and the readers are requested to follow this article for thorough knowledge of SLAM since past two or three decades.

5.2 Modern SLAM System: The GraphSLAM

Full SLAM has been formulated in variety of representation: belief nets, factor graph, Markov random field, Dynamic Bayesian Networks (DBNs), constrained graph.

The first mention of relative, graph-like constraints in the SLAM literature was in (Durrant-Whyte, 1988; Smith and Cheeseman, 1986). However, these authors did not implement any global relaxation or optimization (Thrun and Montemerlo, 2005). The current *de facto* standard formulations of GraphSLAM has its origins in the seminal work of Lu and Milios (1997a), followed by the work of Gutmann and Konolige (2000) and Gutmann and Nebel (1997). They christened it as *GraphSLAM* and represented the SLAM prior as a set of links between robot poses, and formulated a global optimization algorithm for generating a map from such constraints. Gutmann and Nebel (1997) actually implemented the algorithm and reported some numerical instabilities with matrix inversion. Golfarelli, Maio, and Rizzi (1998) proposed a spring-mass model based on (Lu and Milios, 1997a), where knowledge of the environment is represented by using relational graph: the landmarks are the vertices and inter-landmark routes as arcs (see Fig. 5.6).

Duckett, Marsland, and Shapiro (2000) and Duckett, Marsland, and Shapiro (2002) proposed solution to such a spring-mass model. Since then a number of approaches (Dellaert and Kaess, 2006; Folkesson and Christensen, 2004; Grisetti et al., 2007; Kaess et al., 2012; Kaess, Ranganathan, and Dellaert, 2008; Olson, Leonard, and Teller, 2006; Thrun and Montemerlo, 2006; Borrmann et al., 2008a) were proposed improving the efficiency and robustness of the optimization underlying the problem. It should be noted that, Lu and Milios optimization was developed for 2D range scans and optimizes only 3 degrees of freedom (DoF). In this thesis, the extension of (Lu and Milios, 1997a; Lu and Milios, 1997b) to 6 DoF (x, y, z, *roll, pitch, yaw*) by Borrmann et al. (2008a) has been utilized.

5.2.1 Globally Consistent Mapping

Complex 3D digitalization and modelling with no occlusion requires multiple 3D scans. The problem of aligning n partially overlapping scans into a model without inconsistencies is called "*globally consistent scan matching*". A globally consistent map of an environment is a fundamental requirement for the robot localization and navigation. Iterative pairwise matching of individual scans to build complete map is not correct, as it piles up errors from laser scans, the odometry readings and the matching procedure itself. So, global relaxation is necessary to distribute the errors throughout to get consistent representation. Chen and Medioni (1991) introduced an incremental method in which new scans are registered against *meta-scan* to achieve globally consistent range image alignment. However, this approach does not relax the error globally. Krishnan et al. (2005) presented a global registration method that minimized the global-error function by optimization on the manifold of 3D rotation matrices.

In this thesis, 6DSLAM (Borrmann et al., 2008a) has been utilized, which is an extension of Lu and Milios approach from 2D scans to 6DOF. In the next few sections of this chapter, the original global optimization of 2D range scans mapping by Lu and Milios is presented and then its extension to 3D scans is given in detail.

5.2.2 Scan Matching

ICP presented in Chapter 4 is the most used algorithm to match two scans. ICP is used to calculate the transformation between two consecutive scans, as the robot or camera continuously acquires data of the environment. It calculates the optimal $\mathbf{R}, \mathbf{t}$: rotation and translation, between two scans which minimizes (4.4). A straightforward method to 3D reconstruct an environment is to use *pairwise ICP*, which aligns the two consecutive scans, and when *loop closure* is detected, use it to distribute the error globally. In this thesis, all scans are registered sequentially using the ICP algorithm until convergence. The odometry of the new scans is extrapolated to 6 DoF using registration matrices of previously registered scans (the equations solved below are taken from the documentation of Borrmann et al. (2008a) with permission). The change in the pose ΔP is

given as:

$$\begin{pmatrix} x^{\text{odo}}_{n+1} \\ 0 \\ z^{\text{odo}}_{n+1} \\ 0 \\ \theta^{\text{odo}}_{y,n+1} \\ 0 \end{pmatrix} = \begin{pmatrix} x^{\text{odo}}_{n} \\ 0 \\ z^{\text{odo}}_{n} \\ 0 \\ \theta^{\text{odo}}_{y,n} \\ 0 \end{pmatrix} + \left(\begin{array}{c|c} R(\theta_{x,n}, \theta_{y,n}, \theta_{z,n}) & 0 \\ \hline 0 & I_3 \end{array}\right) \cdot \underbrace{\begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta z_{n+1} \\ \Delta\theta_{x,n+1} \\ \Delta\theta_{y,n+1} \\ \Delta\theta_{z,n+1} \end{pmatrix}}_{\Delta P}.$$

Up on Matrix inversion and extracting ΔP, the 6D pose at $(n+1)$ position is then given as:

$$P_{n+1} = \Delta P \cdot P_n$$

where, $(x^{\text{odo}}_n, z^{\text{odo}}_n, \theta^{\text{odo}}_{y,n})$, $(x^{\text{odo}}_{n+1}, z^{\text{odo}}_{n+1}, \theta^{\text{odo}}_{y,n+1})$ are the odometry information of two consecutive robot poses (n and $n+1$ respectively), and $R(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})$ is the registration matrix. It should be noted that the odometry data are in left-handed coordinate system: y represents elevation. Once the distance between poses of two scans falls below a certain threshold, global relaxation is performed using GraphSLAM. For each iteration, a network of pose relations is built automatically. From the corresponding scans, a linear equation system representing distance measurements is built and solved, resulting in optimized pose equations.

5.2.3 Lu and Milios Global Relaxation

Consider a robot traversing a path, with $n+1$ poses $(V_0, \ldots, V_n)$, at each pose making an acquisition. A network of relations is established by matching two scans taken at different positions. A graph is constructed from the nodes $(X_0, \ldots, X_n)$ which are the poses and edges $(D_{i,j})$ being the relations between them. Given, such a graph, the problem is to estimate optimally all the poses to build a consistent map of the environment. For simplification, the measurement equation is assumed to be linear:

$$D_{i,j} = X_i - X_j \tag{5.19}$$

However, the true underlying difference $\bar{D}_{i,j}$ after considering the Gaussian error $\Delta D_{i,j}$ is:

$$\bar{D}_{i,j} = D_{i,j} + \Delta D_{i,j} \tag{5.20}$$

with covariance matrix $C_{i,j}$ assumed to be known.

Maximum likelihood estimates the optimal poses $X - i$, assuming that all the errors in observation are Gaussian and independently distributed, maximizing the probability of all $D_{i,j}$, given their actual observations $\bar{D}_{i,j}$, is equivalent to minimizing the Mahalanobis distance:

$$\mathbf{W} = \sum_{(i,j)} (D_{i,j} - \bar{D}_{i,j})^T C^{-1}_{i,j} (D_{i,j} - \bar{D}_{i,j}) \tag{5.21}$$

which implies

$$\mathbf{W} = \sum_{(0 \le i \le j \le n)} (X_i - X_j - \bar{D}_{i,j})^T C^{-1}_{i,j} (X_i - X_j - \bar{D}_{i,j}) \tag{5.22}$$

with an assumption that network is fully connected and considering simple linear case of estimation problem. In the case of missing link $D_{i,j}$, the corresponding covariance matrix is set to zero. To minimize the above equation, a coordinate system is defined by setting one node (X_0=$(0,0,0)$) as reference point and the other n free nodes $(X_1 \ldots X_n)$ relative to the pose of X_0. Using the signed incidence matrix $\mathbf{H}$, the concatenated measurement equation $\mathbf{D}$ is:

$$\mathbf{D} = \mathbf{H}\mathbf{X}, \tag{5.23}$$

that makes,

$$\mathbf{W} = \left(\bar{\mathbf{D}} - \mathbf{HX}\right)^T \mathbf{C}^{-1} \left(\bar{\mathbf{D}} - \mathbf{HX}\right) \tag{5.24}$$

The concatenations of all the observations $\bar{D}_(i,j)$ forms the vector $\bar{\mathbf{D}}$, while $\mathbf{C}$ is a block-diagonal matrix composed of $C_{i,j}$ as sub-matrices. The solution $\mathbf{X}$ that minimizes (5.22) and its covariance matrix $\mathbf{C_X}$ are given by:

$$\mathbf{X} = \left(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{C}^{-1}\bar{\mathbf{D}} \text{ and } \mathbf{C_X} = \left(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H}\right)^{-1} \tag{5.25}$$

For simple representations, $\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H}$ as $\mathbf{G}$ and $\mathbf{H}^T\mathbf{C}^{-1}\bar{\mathbf{D}}$ as $\mathbf{B}$ makes:

$$\mathbf{GX} = \mathbf{B} \tag{5.26}$$

where,

$$G_{i,j} = \begin{cases} \sum_{j=0}^{n} C_{i,j}^{-1} & (i = j) \\ C_{i,j}^{-1} & (i \neq j). \end{cases} \tag{5.27}$$

The entries of $\mathbf{B}$ are obtained by:

$$B_i = \sum_{\substack{j=0 \\ j\neq i}}^{n} C_{i,j}^{-1}\bar{D}_{i,j}. \tag{5.28}$$

5.2.4 Extension to 6 DoF

Extending to 6 DoF, assume that a robot starts at the pose $V_b = (x_b, y_b, z_b, \theta_{x_b}, \theta_{y_b}, \theta_{z_b})^T$ and changes its pose by $D = (x, y, z, \theta_x, \theta_y, \theta_z)^T$ relative to V_b, ending up at $V_a = (x_a, y_a, z_a, \theta_{x_a}, \theta_{y_a}, \theta_{z_a})^T$. The poses V_a and V_b are related by the compounding operation $V_a = V_b \oplus D$. Similarly, a 3D position vector $u = (x_u, y_u, z_u)$ is compounded with the pose V_b by $u' = V_b \oplus u$:

$$\begin{aligned} x'_u &= x_b - z_u \sin\theta_{y_b} + \cos\theta_{y_b}(x_u\cos\theta_{z_b} - y_u\sin\theta_{z_b}) \\ y'_u &= y_b + z_u\cos\theta_{y_b}\sin\theta_{x_b} + \cos\theta_{x_b}(y_u\cos\theta_{z_b} + x_u\sin\theta_{z_b}) \\ &\quad + \sin\theta_{x_b}\sin\theta_{y_b}(x_u\cos\theta_{z_b} - y_u\sin\theta_{z_b}) \\ z'_u &= z_b - \sin\theta_{x_b}(y_u\cos\theta_{z_b} + x_u\sin\theta_{z_b}) \\ &\quad + \cos\theta_{x_b}(z_u\cos\theta_{y_b} + \sin\theta_{y_b}(x_u\cos\theta_{z_b} - y_u\sin\theta_{z_b})) \end{aligned}$$

This operation is used to transform a non-oriented point from its local to the global coordinate system.

Scan matching computes a set of m corresponding point pairs u_k^a, u_k^b between two scans, each representing a single physical point. The positional error made by identifying these two points in different scans is described by:

$$F_{ab}(V_a, V_b) = \sum_{k=1}^{m} \left\| V_a \oplus u_k^a - V_b \oplus u_k^b \right\|^2 \tag{5.29}$$

$$= \sum_{k=1}^{m} \left\| (V_a \ominus V_b) \oplus u_k^a - u_k^b \right\|^2. \tag{5.30}$$

Based on these m point pairs, the algorithm computes the matrices $\bar{D}_{i,j}$ and $C_{i,j}$ for solving (5.22). $\bar{D}_{i,j}$ is derived as follows.

Let $\bar{V}_a = (\bar{x}_a, \bar{y}_a, \bar{z}_a, \bar{\theta}_{x_a}, \bar{\theta}_{y_a}, \bar{\theta}_{z_a})$ and $\bar{V}_b = (\bar{x}_b, \bar{y}_b, \bar{z}_b, \bar{\theta}_{x_b}, \bar{\theta}_{y_b}, \bar{\theta}_{z_b})$ be close estimates of V_a and V_b. If the global coordinates of a pair of matching points $u_k = (x_k, y_k, z_k)$, then (u_k^a, u_k^b) fulfill the equation:

$$u_k \approx V_a \oplus u_k^a \approx V_b \oplus u_k^b.$$

For small errors $\Delta V_a = \bar{V}_a - V_a$ and $\Delta V_b = \bar{V}_b - V_b$, a Taylor expansion leads to:

$$\begin{aligned}
\Delta Z_k &= V_a \oplus u_k^a - V_b \oplus u_k^b := F_k(V_a, V_b) \\
&\approx F_k(\bar{V}_a, \bar{V}_b) - [\nabla_{\bar{V}_a}(F_k(\bar{V}_a, \bar{V}_b))\Delta V_a \\
&\quad - \nabla_{\bar{V}_b}(F_k(\bar{V}_a, \bar{V}_b))\Delta V_b] \\
&= \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b - [\nabla_{\bar{V}_a}(\bar{V}_a \oplus u_k^a)\Delta V_a \\
&\quad - \nabla_{\bar{V}_b}(\bar{V}_b \oplus u_k^b)\Delta V_b]
\end{aligned} \tag{5.31}$$

where $\nabla_{\bar{V}_a}(F_k(\bar{V}_a, \bar{V}_b))$ is the gradient of the pose compounding operation. By matrix decomposition

$$\begin{aligned}
M_k H_a &= \nabla_{\bar{V}_a}(F_k(\bar{V}_a, \bar{V}_b)) \\
M_k H_b &= \nabla_{\bar{V}_b}(F_k(\bar{V}_a, \bar{V}_b)),
\end{aligned}$$

and (5.31) simplifies to:

$$\begin{aligned}
\Delta Z_k &\approx \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b - M_k[H_a \Delta V_a - H_b \Delta V_b] \\
&= \bar{Z}_k - M_k D
\end{aligned}$$

with

$$\begin{aligned}
\bar{Z}_k &= \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b \\
D &= (H_a \Delta V_a - H_b \Delta V_b) \\
M_k &= \begin{pmatrix} 1 & 0 & 0 & 0 & -y_k & -z_k \\ 0 & 1 & 0 & z_k & x_k & 0 \\ 0 & 0 & 1 & -y_k & 0 & x_k \end{pmatrix}
\end{aligned} \tag{5.32}$$

$$H_a = \begin{pmatrix} 1 & 0 & 0 & 0 & \bar{z}_a \cos(\bar{\theta}_{x_a}) + \bar{y}_a \sin(\bar{\theta}_{x_a}) & \bar{y}_a \cos(\bar{\theta}_{x_a}) \cos(\bar{\theta}_{y_a}) - \bar{z}_a \cos(\bar{\theta}_{y_a}) \sin(\bar{\theta}_{x_a}) \\ 0 & 1 & 0 & -\bar{z}_a & -\bar{x}_a \sin(\bar{\theta}_{x_a}) & -\bar{x}_a \cos(\bar{\theta}_{x_a}) \cos(\bar{\theta}_{y_a}) - \bar{z}_a \sin(\bar{\theta}_{y_a}) \\ 0 & 0 & 1 & \bar{y}_a & -\bar{x}_a \cos(\bar{\theta}_{x_a}) & \bar{x}_a \cos(\bar{\theta}_{y_a}) \sin(\bar{\theta}_{x_a}) + \bar{y}_a \sin(\bar{\theta}_{y_a}) \\ 0 & 0 & 0 & 1 & 0 & \sin(\bar{\theta}_{y_a}) \\ 0 & 0 & 0 & 0 & \sin(\bar{\theta}_{x_a}) & \cos(\bar{\theta}_{x_a}) \cos(\bar{\theta}_{y_a}) \\ 0 & 0 & 0 & 0 & \cos(\bar{\theta}_{x_a}) & -\cos(\bar{\theta}_{y_a}) \sin(\bar{\theta}_{x_a}) \end{pmatrix}.$$

H_b is given analogously. This matrix decomposition and the derivation of H_a, H_b is the crucial step in extending Lu and Milios style SLAM to 6 DoF.

D as defined by (5.32) is the new linearized measurement equation. To calculate both $\bar{D}$ and C_D, (5.30) is rewritten in matrix form:

$$F_{ab}(D) \approx (\mathbf{Z} - \mathbf{M}D)^T (\mathbf{Z} - \mathbf{M}D).$$

$\mathbf{M}$ is the concatenated matrix consisting of all M_k's, and $\mathbf{Z}$ the concatenated vector consisting of all Z_k's. The vector $\bar{D}$ that minimizes F_{ab} is given by

$$\bar{D} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{Z}. \tag{5.33}$$

Since minimizing F_{ab} constitutes least squares linear regression, the Gaussian distribution of the solution is modelled with mean $\bar{D}$ and standard covariance estimation

$$C_D = s^2(\mathbf{M}^T\mathbf{M}). \tag{5.34}$$

s^2 is the unbiased estimate of the covariance of the identically, independently distributed errors of Z_k, given by:

$$s^2 = (\mathbf{Z} - \mathbf{M}\bar{D})^T(\mathbf{Z} - \mathbf{M}\bar{D})/(2m-3) = \frac{F_{ab}(\bar{D})}{2m-3}.$$

The error term W_{ab} corresponding to our pose relation is defined by:

$$W_{ab} = (\bar{D} - D)^T C_D^{-1}(\bar{D} - D).$$

5.2.4.1 Transforming the Solution

Solving the linear equation (5.26) leads to an optimal estimate of the new measurement equation of D (5.32). To yield an optimal estimation of the robot poses, it is necessary to transform D. By this optimal estimation, a set of solutions $X_i = H_i\Delta V_i$ is computed, each corresponding to a node in the network. Assuming that the reference pose $V_0 = 0$, the pose V_i and its covariance C_i are updated by:

$$V_i = \bar{V}_i - H_i^{-1}X_i,$$
$$C_i = (H_i^{-1})C_i^X(H_i^{-1})^T.$$

If V_0 is nonzero, the solutions have to be transformed by:

$$V_i' = V_0 \oplus V_i$$
$$C_i' = K_0 C_i K_0^T$$

where

$$K_0 = \begin{pmatrix} R_{\theta_{x_0},\theta_{y_0},\theta_{z_0}} & 0 \\ 0 & I_3 \end{pmatrix}$$

with a rotation matrix $R_{\theta_{x_0},\theta_{y_0},\theta_{z_0}}$.

5.2.4.2 The Algorithm

The optimal estimation algorithm is given as Algorithm 3. Iterative execution of Algorithm 3 yields a successive improvement of the global pose estimation. Step 3 is sped up by component-wise computation of $\mathbf{G}$ and $\mathbf{B}$. The components $C_{i,j}^{-1} = (\mathbf{M}^T\mathbf{M})/s^2$ and $C_{i,j}^{-1}\bar{D}_{i,j} = (\mathbf{M}^T\mathbf{Z})/s^2$ are expanded into simple summations. The most expensive operation is solving the linear equation system $\mathbf{GX} = \mathbf{B}$. Since $\mathbf{G}$ is a positive definite, symmetric $6n \times 6n$ matrix, this is done by Cholesky decomposition in $\mathcal{O}(n^3)$.

5.3 Datasets

Publicly available and benchmark datasets help to push forward the state-of-the-art techniques in Computer Vision, Image Processing, Machine Learning, Robotics and several other scientific domains. They support the scientific evaluation and objective comparison of algorithms with a clear evaluation metrics. SLAM is one of the problems in robotics which has been investigated using a variety of image and time-of-flight sensors that use radar, sonar and LiDAR (Cadena

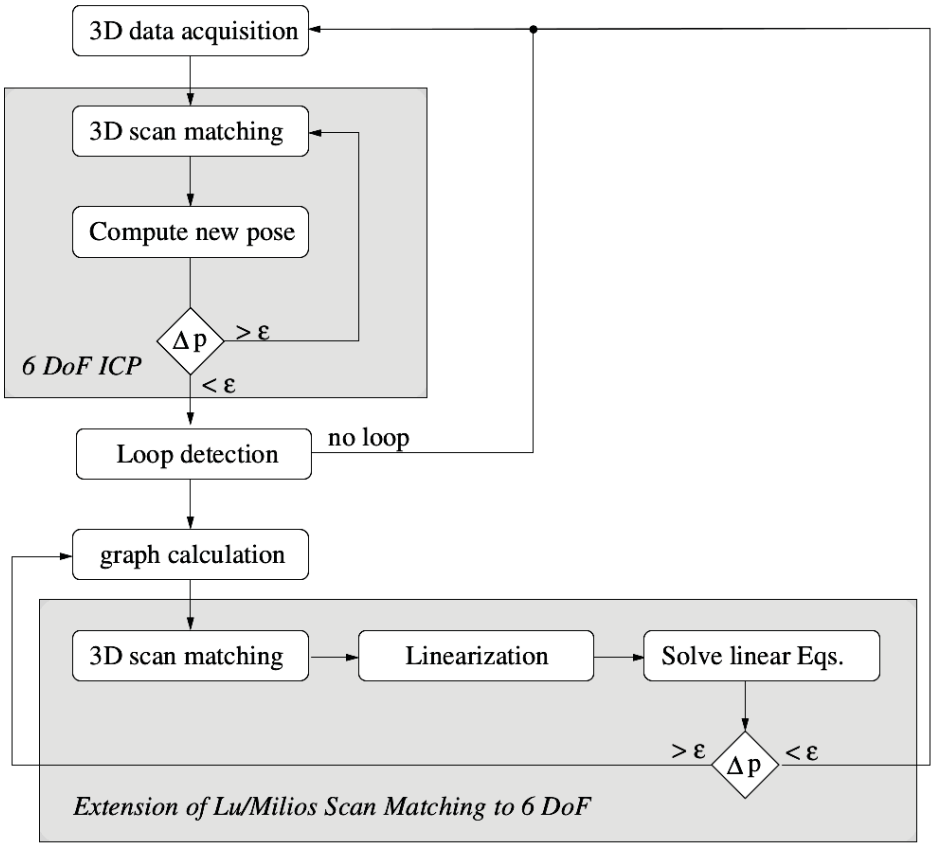


Figure 5.7 – Globally consistent 3D mapping, an extension to 6 DoF. Source: (Borrmann et al., 2008a).

Algorithm 3 Optimal estimation algorithm

1. Compute the point correspondences u_k^a, u_k^b.
2. For any link (i, j) in the given graph compute the measurement vector $\bar{D}_{ij}$ by (5.33) and its covariance C_{ij} by (5.34).
3. From all $\bar{D}_{ij}$ and C_{ij} form the linear system $\mathbf{GX} = \mathbf{B}$, with $\mathbf{G}$ and $\mathbf{B}$ as given in (5.27) and (5.28) respectively.
4. Solve for $\mathbf{X}$
5. Update the poses and their covariances, as explained in Section 5.2.4.1.

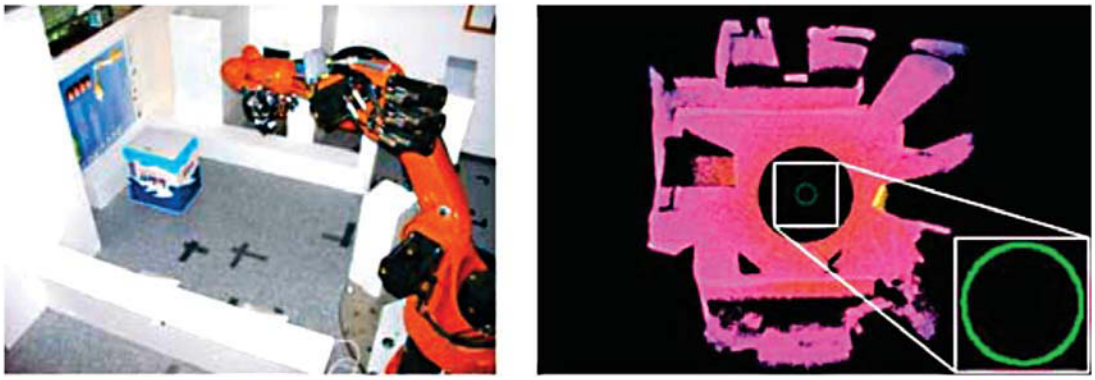


Figure 5.8 – Mapping dataset made using SwissRanger time-of-flight camera mounted on KUKA effector. Source: (Stefan Fuchs, 2017).

et al., 2016). A good source of SLAM dataset acquired with different types of sensors can be found at (Andreas Nüchter and Kai Lingemann, 2017). Since the last decade, cameras that use the time-of-flight technique have become ubiquitous in robotic applications. They are commercially cheap, simple and provide data at video frame rates. The first SLAM systems that used RGB-D sensors have appeared in (Henry et al., 2014; Dib, Beaufort, and Charpillet, 2014; Comport, Meill, and Rives, 2011), though these sensors use *structured light* rather than the time-of-flight technique. Sturm et al. (2012) presented a RGB-D benchmark dataset for SLAM using Kinect (v 1.0) RGB-D camera. However, presently, the data from the even best RGB-D camera (Kinect 2.0, Asus Xtion PRO live) is extremely noisy with limited maximum range (4.5 m) and larger measurement accuracy ($\pm$ 0.03 m for 3 m range). And for mapping with these sensors the data should be acquired within 13 m distance from the surroundings (Khoshelham and Elberink, 2012). On the other hand, SwissRanger time-of-flight camera is although noisy (with measurement accuracy $\pm$ 0.01 m for 5 m & 10 m range) but has better and well studied noise characteristics (Weingarten, Gruener, and Siegwart, 2004; Dopfer, Wang, and Wang, 2014; Tamas and Jensen, 2014; Chiabrando et al., 2009; Donoho, 1995; Jovanov, Pižurica, and Philips, 2010; He et al., 2017; Diebel and Thrun, 2005; Cazorla, Viejo, and Pomares, 2010; Falie and Buzuloiu, 2007; Foix, Alenya, and Torras, 2011a; Hansard et al., 2012; Lange, 2000; Robbins et al., 2008; Kahlmann, Remondino, and Ingensand, 2006; Ghorpade, Checchin, and Trassoudaine, 2015; Reynolds et al., 2011) with a maximum range of 10 m and higher frequency than Kinect, but without RGB channel instead provides confidence and amplitude images. Also, the data from it are much smoother than Kinect like sensors (as it uses time-of-flight principle not structured light, unlike Kinect). The phase shift principle of it helps to acquire data at longer range more accurately than Kinect. Well studied noise characteristics of SwissRanger camera helps to remove both systematic and non-systematic errors at ease.

Future service robots largely depend on time-of-flight cameras for mapping, navigation, manipulation and semantic interpretation of their surroundings and will certainly use only the robust, light-weighted and simple sensors for this purpose. SwissRanger time-of-flight camera full-fill these criteria compared to Kinect or any other state-of-the-art sensor presently available (Hong et al., 2012; Ye and Bruch, 2010; Cui et al., 2010; Kolb et al., 2009; May et al., 2009b; May et al., 2009a; Fuchs and May, 2008; Chiabrando, Piatti, and Rinaudo, 2010; Iddan and Yahav, 2001) (see Fig. 3.3 for present state-of-the-art depth sensors). Moreover, they are improved constantly when compared to Kinect family of sensors which are explicitly developed for video-gaming not for robotics.

In this thesis, a standard dataset from time-of-flight camera is made to experiment and evaluate SLAM algorithms, and it is publicly available[4]. The pose information of the camera at every acquisition position is also provided. To the best of our knowledge, colorless point cloud

[4] ftp://ftp.ip.univ-bpclermont.fr/iptof-d

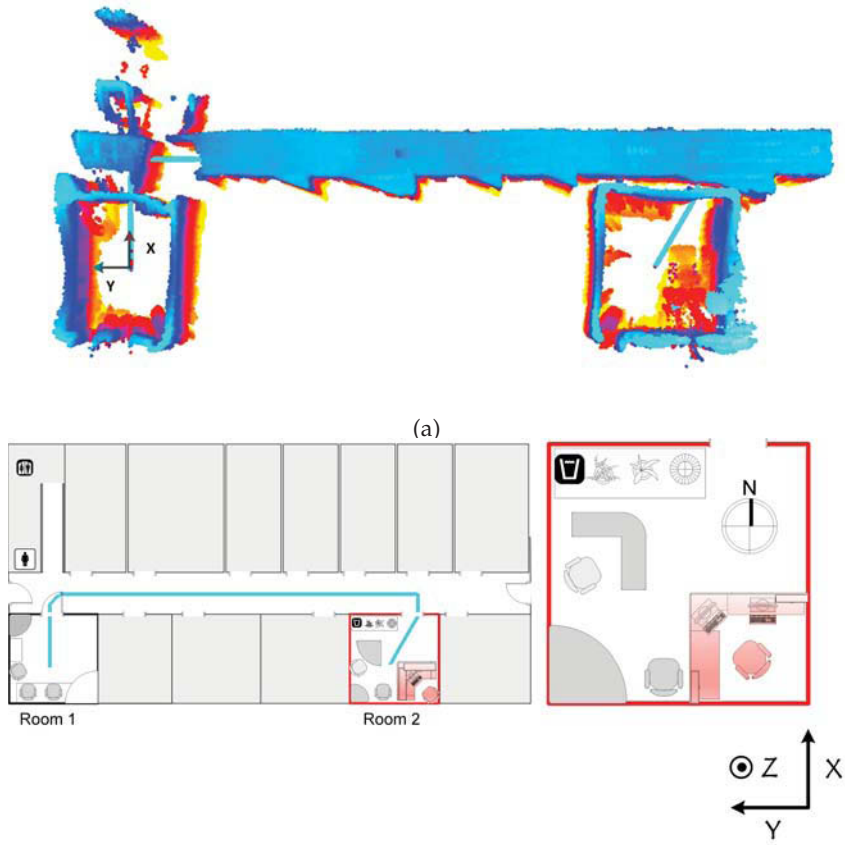


Figure 5.9 – Institut Pascal Semantic SLAM dataset. (**a**) Map of the indoor environment, two offices connected with a long corridor. The dataset consists of 484 scans taken at pre-arranged locations. The map is reconstructed using 3DTK (Borrmann et al., 2008a), the heat-map representation is proportional to the distance above the ground. (**b**) The path of the indoor environment across which scans are acquired. The pose files have the x, y, z, θ_z information. A floor plan of the environment is overlaid on the path for reference only. Scale and representation are not accurate. Chairs and tables are positioned in the two rooms to serve as objects for semantic interpretation. Several household objects are placed on the desk in the right room (towards the north, see Fig. 1.5). These are the same objects represented in (c), for semantic labelling of the scene. (**c**) Different household objects have been placed in the indoor environment to be labelled after the 3D mapping. The object dataset consists of 2.5D views of different objects, taken at