DISS. ETH. NO 21159

APPLIED REGISTRATION FOR ROBOTICS
Methodology and Tools for ICP-like Algorithms

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

FRANÇOIS POMERLEAU

MSc Université de Sherbrooke, Electrical Engineering

born November 20, 1981

citizen of Canada

accepted on the recommendation of

Prof. Roland Y. Siegwart (ETH Zurich, Principal Advisor)

Dr. Mike Bosse (Senior Research Scientist, CSIRO, Australia, Member of the Jury)

2013

Acknowledgements

Multiple people invested themselves in the accomplishment of this thesis. My warmest gratitude goes to my supervisor Professor Roland Siegwart. He gave me the opportunity to enjoy my first field experimentation with an autonomous robot and trusted my capabilities in Robotics from EPFL to ETH. As head of the Autonomous Systems Lab, he opened opportunities to concretize my ideas. Roland was a great mentor with his supportive attitude and his excitements for new developments.

I'm also specially thankfully to Francis Colas, Stéphane Magnenat and Cédric Pradalier for their constant support. From the beginning, Francis patient explanations guided me through better mathematical foundations. I really appreciated his help during the review of this thesis and his dedication in making the project NIFTi a success. Stéphane shared with me his skillful talents in programming and his eagerness for open source software. Over the years, Cédric provided me research directions and orientated toward precious resources. He was also always willing for a short brainstorming or a coding session over nights.

Maintaining a robotic platform can be tedious and required a lot of devotions. Therefor, I wish to thank Ming Liu and Francis for their work on NiftiBot, in collaboration with the complete NIFTi consortium; Fabien Tâche and Andreas Breitenmoser, who gave my the opportunity to work with the Magnebike; Gregory Hitz for his long time passed outdoor, working the boat Lizhbeth; Philipp Krüsi for his continuous development on Artor; and Rudolph Triebel and Luciano Spinello leading one of the latest data recording on the SmartTer.

Student supervision helped to explore a broader range of ideas. Their contributions to my research were very appreciated. Specially Elena Stumm for her work on path planning; Gregory Hitz, Manuel Baumann, Oliver Baur for the development of an autonomous surface vessel; and Benoît Lescot for his supportive work on data sets recording.

Financial supports are always needed to conduct research. I want to mark my appreciation for the fellowship received from the Fonds québécois de recherche sur la nature et les technologies (FQRNT), which allowed me to move to Zurich, Robots@home (FP6-IST-045350) and NIFTi (FP7-ICT-47870) which make me travelled and meet interesting people through Europe. Of course the support for electronic and mechanic during research implying robots, and therefor I gracefully thanks Stefan, Thomas, Markus and Dario for sharing there talents. Also all my thanks and acknowledgement to Lucy and Cornelia for their administrative assistance.

Special thanks to my friends and colleges: Cédric, Stéphane, Francis keeper of the chocolate, Ambroise, Fabien, Xavier, Gilles (for the grappa), Ming (for all the games played during the long travels through Europe), Dizan, the people of the Limnological Station for the relaxing time by the lake, Jérôme and Ralf (for including sport in my schedule), ASL team for the BBQs on the roof, movie nights, beers in the BQM, games in the KoZe and all other activities spicing up the research life. To my family, which were patient during those times abroad.

And finally, to Marie-Ève for being so joyful and supportive during those years. Her special attentions during the redaction made this thesis possible.

Abstract

Registration algorithms allow to associate two representations of a given environment in a same reference frame. Although algorithms for image registrations are very useful, we focused our effort toward geometric registration. It is usually the foundation of 2D and 3D mapping and an important competence for a mobile robot to have. Solutions were found 20 years ago, namely with the publication of the well-known Iterative Closest Point (ICP) algorithm, but the number of publications on the topic kept growing, even at a faster rate in the recent years. This situation underlies a stronger problem related to the methodology used in robotics.

The simplicity of the ICP makes it an attractive solution for researchers looking for an algorithm allowing them to test higher-level tasks (e.g., path planning, exploration, etc.). Naively implemented, its simplicity also lead to a number of flaws appearing when used in critical applications where an autonomous system needs to take decisions in an unknown environment. This explains why many variations of the original solution, adapted in multiple ways, are published every year. Specially in robotics, the number of possible adaptations becomes so large that it is hard to take clever decisions about which algorithm should be implemented on a robot.

We addressed this problematic by providing a more structured methodology to evaluate geometric registrations for robotic applications. To do so, we surveyed a large number of solutions published and we related them together in a unique framework. We reused this framework to design and implement a registration library to compare the performance of different solutions in different conditions. We structured an evaluation protocol around eight data sets recorded over a period of half a year, with ground truth positioning precision in the order of millimeters. Those data sets cover a variety of environmental structures, as they were recorded indoors and outdoors and include multiple types of dynamic elements.

The computational performance of the proposed library was demonstrated by handling geometric registration in scenarios where real-time processing is needed. The evaluation of standard ICP solutions using our proposed methodology highlighted interesting and varying behaviors in different conditions. This allowed us to rapidly tune tailored registration algorithms for five robotic platforms. We used successfully those platforms as case studies covering mobile systems on ground, water and air, with different motion capabilities and sensors.

Keywords: Geometric Registration, Iterative Closest Point, ICP, 3D Mapping, Data Sets, Methodology, Laser Rangefinder, Robotics

Résumé

Les algorithmes de registration permettent d'associer deux représentations d'un environnement donné dans le même plan. Bien que les algorithmes de registration d'images soient très utiles, nos efforts se sont concentrés sur la registration géométrique. Ce type de registration constitue généralement les fondements de la cartographie 2D et 3D, et est l'une compétence essentielle que doit posséder un robot mobile. Plusieurs solutions ont été trouvées depuis 20 ans, particulièrement suite à la publication du célèbre algorithme ICP, mais le nombre de publications continue toujours d'augmenter, voire même à un rythme accéléré ces dernières années. Cette situation sous-tend un problème plus grave, qui est relié à la méthodologie utilisée en robotique.

La simplicité de l'ICP en fait une solution attrayante pour les chercheurs désirant un algorithme qui leur permet de tester des tâches complexes (planification de chemins, exploration, etc.). Lorsque ingénument implémenté, sa simplicité peut aussi amener des failles qui peuvent apparaître dans une situation critique où le système autonome doit prendre des décisions dans un environnement inconnu. C'est pourquoi de nombreuses variations de la solution originale, adaptée de multiple façons, sont publiées chaque année. Spécialement en robotique, le nombre d'adaptations possibles est devenu si grand qu'il devient difficile de prendre des décisions éclairées au sujet de l'algorithme qui devrait être implémenté sur un robot.

Nous avons abordé cette problématique en fournissant une méthodologie davantage structurée afin d'évaluer la registration géométrique lors d'applications en robotique. Pour ce faire, nous avons passé en revue un grand nombre de solutions publiées, et nous les avons réunies dans un cadre unique. Nous avons réutilisé ce cadre pour concevoir et implémenter une bibliothèque de registrations afin de comparer la performance de différentes solutions dans diverses conditions. Nous avons élaboré un protocole d'évaluation à partir de huit jeux de données qui furent enregistrés au cours d'une période de 6 mois, et ce, avec une précision de positionnement de l'ordre du millimètre lorsque comparé à la réalité de terrain. Ces jeux de données couvrent une variété de structures environnementales puisqu'ils ont été enregistrés tant à l'intérieur qu'à l'extérieur, et incluent de multiples genres d'éléments dynamiques.

La performance informatique de la bibliothèque proposée a été démontrée lors du traitement de registrations géométriques dans des scénarios où le traitement en temps réel est nécessaire. L'évaluation de solutions standard de l'ICP avec notre méthodologie a mis en lumière des comportements intéressants et variants selon les différentes conditions. Cela nous a permis d'ajuster rapidement les algorithmes de registrations pour cinq plateformes robotisées. Nous avons utilisé avec succès ces plateformes comme études de cas couvrant des systèmes mobiles sur le sol, dans l'eau et dans l'air, et ayant différentes capacités de mouvements et possédant divers capteurs.

Mots-clés: Registration géométrique, *Iterative Closest Point* - Point rapproché itératif, ICP, Cartographie 3D, Jeux de données, Méthodologie, Télémètre à laser, Robotique

Contents

Chapter 1

Twenty years of ICP: The Legacy

The scope of this thesis tackles the complex topic of autonomous 3D reconstruction using *registration* algorithms. Registration algorithms associate sets of data into a common coordinate system by minimizing the alignment error. Although they can be quite an abstract concept, registration solutions already had an impact on the artistic field and popular culture. Photographers proficiently use image registration to build photograph composites achieving different looks-and-feels. The Brenizer method is an exemplary technique that is applied to achieve dramatic depth of field using panoramic image stitching (Figure 1.1 - left). Another example is High Dynamic Range (HDR) photographs, where multiple images at different exposure levels need to be precisely overlaid to retrieve details in shaded and highlighted areas (Figure 1.1 - right). Nowadays, even the latest cellphones have the capacity to build panoramic images from a series of pictures taken based on a visual guidelines that direct the user to move the camera viewfinder at the optimal position for the next picture. Even the specific case of 3D mapping application had inspired cinematographers, which had depicted possible uses of registration algorithms in several recent science fiction movies. For instance, in the remake of *Total Recall* (Colombia Pictures, 2012), an armed intervention team employed an array of hundreds of tiny cameras in a dangerous room leading to a 3D reconstruction of the area used to monitor potential threats within couple of seconds. Another closely related potential application was the used by a geologist of flying drones carrying laser rangefinders to explore an alien facility in *Prometheus* (Twentieth Century Fox, 2012).

Figure 1.1: Example of image registrations used in photography. *Left*: Brenizer method using the open source software Hugin to stitch multiple images. *Right*: HDR composite of the San Francisco harbor using the open source software Luminance HDR to overlay three images.

More at the research level, current applications include: robotic exploration in harsh environments, organ reconstruction to improve medical diagnostics and object reconstructions for reverse engineering. Although registration using 2D images can be part of the same group of solutions, we focus on systems where depth information is already available (e.g., from laser rangefinders). We refer to the latter type as *geometric registration*. However, some parallels with image registrations will be made throughout this work when relevant. A simplified example of geometric registration is illustrated in Figure 1.2. A scene with a large tree, a lamppost and a bench was scanned using a laser rangefinder, so only the locations of the objects are available to resolve the alignment error. Although all individual points are similar, their proximity to other points gives enough information to automatically align two consecutive point clouds.

In general, image registrations often have access to *labelled* points, which is less the case for geometric registrations, either in 2D or 3D.

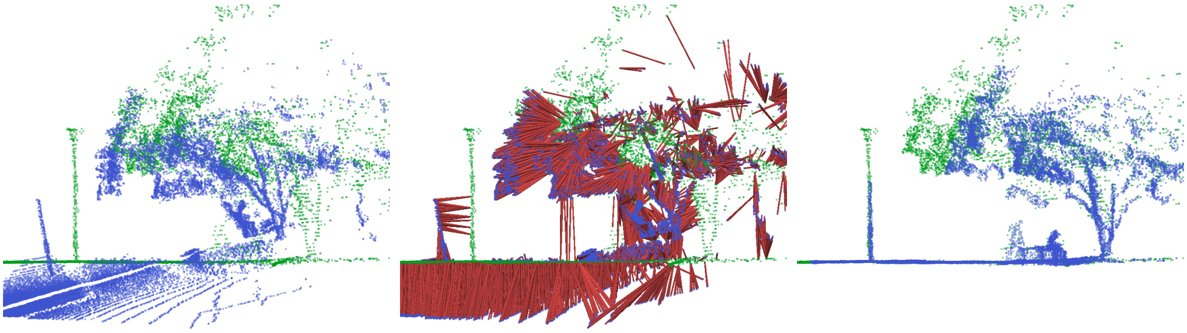


Figure 1.2: Examples of geometric registration between a reference point cloud (green) and a reading point cloud (blue). *Left*: Initial position of the two point clouds. *Middle*: Alignment error (red). *Right*: Final alignment of the two point clouds.

1.1 Algorithm Overview

A general scheme for geometric registration solutions is presented in Figure 1.3. More formally stated, the goal of data association is to be able to represent a geometric set of features that we named `reading` in the same coordinate frame as another set named `reference`. Each feature can be already linked to a descriptor list, which are respectively represented by blue and red (dashed) arrows. The use of the terms features and descriptors is often misleading, so we specify here the nomenclature:

Features include all geometric parameters used to define shapes and susceptible to be affected by a space transformation (e.g., points coordinates, surface normal vectors, tangent vectors, etc.).

Descriptors include all information assumed to be invariant of transformations applied to a reference frame (color, temperature, names, utilities, etc.).

An `initial transformation` is created to represent the `reading` features in the same coordinate frame as the `reference` frame. The `reading` and the `reference` generally come directly from a sensor or from the computer's memory (such as map, atlas, etc.). The `data filters` aim at removing the features carrying a low level of information and at increasing the discrepancy between the remaining features by generating new descriptors. The `match function` associates, based on some assumptions, features and/or descriptors coming from the `reading` and the `reference` together. The `outlier filters` evaluate wrong associations and try to reduce their impacts on the solution. Since descriptors are linked to features, `outlier filters` that are applied to descriptors can be directly transfered in the feature space. This relation is represented by the black arrow linking both `outlier filters`. Finally, an `error minimization` function, that also considers some assumptions, is applied in the feature space to reduce the error created by the `initial transformation`. The main output of the algorithm is a list of parameters, the `final transformation`, allowing to represent the `reading` features in the same coordinate frame as the `reference` features. Dealing with unlabeled points leads to approximations in the `match function` resulting in a non-optimal `final transformation` (i.e, some misalignment errors remain). In those cases, an iterative version of the scheme is used, where the `final transformation` is fed back in a loop to the `initial transformation` up to convergence. This type of solution is known as the Iterative Closest Point (ICP).

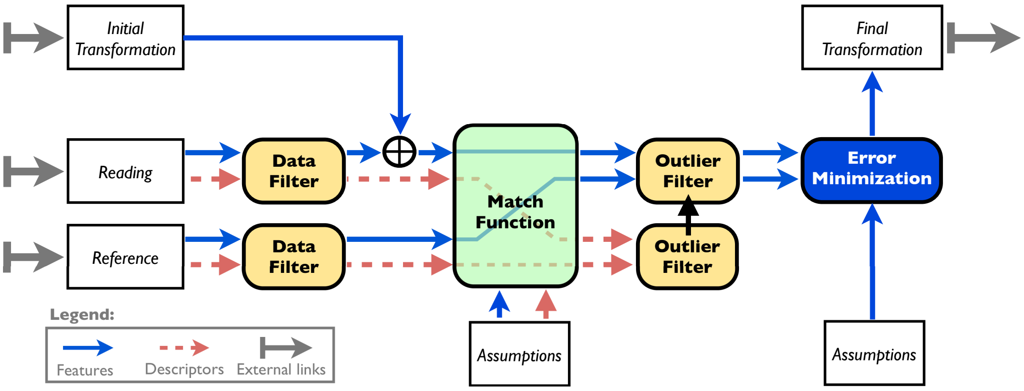


Figure 1.3: Generic scheme proposed for registration algorithms.

1.2 Early Solutions

As an interesting historical note, in an early publication by Hurley and Cattell [1962], registration is presented as an *Orthogonal Procrustes* problem. The name Procrustes refers to a bandit from the Greek mythology who made his victims fit in his bed by either stretching their limbs or cutting them off. Theseus eventually defeated Procrustes using the same violent procedure (Figure 1.4). The reference to the *Orthogonal Procrustes* problem is not often used in the scientific literature, but it illustrates well the idea.

While working more specifically on 3D-shape primitives, Faugeras and Hebert [1986] defined closed-form distances to minimize point-to-point and

Figure 1.4: Theseus *adjusting* Procrustes to the size of his bed. Photograph provided by Marie-Lan Nguyen / Wikimedia Commons.

plane-to-plane alignment error. The proposed method solved translation and rotation as a two-step procedure. Later, a solution proposed by Walker et al. [1991] resolved together rotation and translation error using dual quaternions. The registration problem concretizes itself further in a survey of geometric matching algorithms and geometric representations for point sets, curves, surfaces, volumes, and their respective space-time trajectories [Besl 1988]. At this time, the main validation was using Computer-aided design (CAD) models with simple shapes. The first mention of the name ICP was proposed by Besl and McKay [1992]. They expressed the problem as follows:

> "Given 3-D 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."

In their work, the proof of the solution convergence is demonstrated under the assumption that the number of associated points, or their weight, remains constant. Unless two identical shapes are registered together, outliers that are not present in both shapes need to be identified. This problems is observed by Champleboux et al. [1992] while developing early registration solutions for medical applications. They report failures when wrong initial transformations are used in combination with scans having low overlap ratio. During the same years, Chen and Medioni [1991] work with dense laser scans of status and tooth mockups [Chen and Medioni 1992]. They propose a registration solution based on the minimization of point-to-plane alignment errors, which is still quite used nowadays.

Even though a large volume of theoretical works was published on geometric primitives, Zhang [1994] states that precise primitives based on geometric characteristics are too sensible to noise and are not stable in moving systems with current (1994) sensing capabilities. Thus, he concludes that points were more reliable. Zhang [1994] pioneeres the idea of using ICP-based solutions for outdoor robotic applications. He proposes a generic framework for symmetric match, but considers only one direction of registration as an approximation to save computation costs. He highly emphasizes the importance of removing spurious pairs and gives the first characterization of fast subsampling solutions. In addition, he highlights the fact that outlier rejection is required for robotic applications, and that the proof of ICP convergence stated by Besl and McKay [1992] cannot not hold for most of the applications. In the outlook section of his work, he already mentions the use of uncertainty on the initial alignment, based on Kalman filters and Mahalanobis distance, and the need to handle dynamic elements.

1.3 Division and Explosion of the Field

Within only two years, four main application types already emerged from the possibilities to register 3D point clouds: object reconstructions [Chen and Medioni 1991], non-contact inspections [Besl and McKay 1992], medical and surgery support Chambleboux et al. [1992] and autonomous vehicle navigation [Zhang 1993]. Publications in specialized journals for computer vision, robotics and medical imaging slowly divided the types of *interesting* problems to be solved. We can still read in current literature that the credits for being the first article to provide a solution differ from authors in different fields.

The field of registration crystalized with its first survey on medical image registration covering the years 1993 to 1998 [Maintz and Viergever 1998]. It took 12 years for a specialized survey of 3D registration in Computer Vision to appear [Bowyer et al. 2006], and one in robotics is still to come.

ICP is a popular algorithm due to its simplicity: its general idea is easy to understand and to implement. However, the basic algorithm only works well in ideal cases. This led to hundreds of variations (close to 400 papers based on IEEE Xplore and around 1000 based on Scopus, in the last 20 years, see Figure 1.5) around the original algorithm that was demonstrated on different and numerous number of experimental scenarios. This highlights both the usefulness of ICP and the difficulty to find a single versatile version.

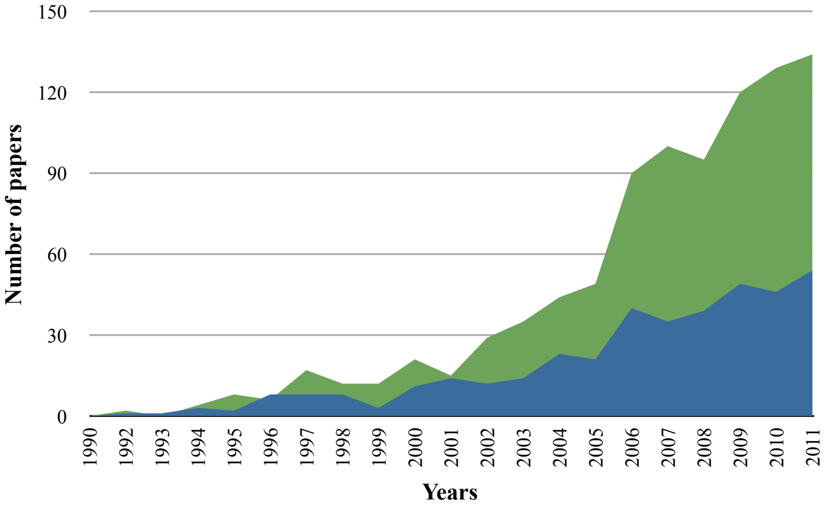


Figure 1.5: Evolution of the number of publications over the years. Results were obtained for the keywords *Iterative Closest Point* appearing in the abstract or the title of publications. The blue area is computed based on IEEE Xplore database and the green area from Scopus.

Figure 1.5 shows an increasing number of publications appearing around the year 2000. In robotics, this coincides with the advent of a 2D solution for pose estimations demonstrated with a SICK rangefinder [Lu and Milios 1997] and of the basis of Simultaneous Localization and Mapping (SLAM) algorithms Thrun et al. [1998]. Prior to the arrival of the SICK LMS-200 in robotics [Pfister et al. 2002], most of the sensors used were custom-made and thus, difficult to replicate by other researchers. In those years, 2D lasers finally appeared as a viable solution for robot navigation over sonars that were traditionally used [Thrun et al. 1998]. The 3D applications were then not accessible due to high computation costs. At the same time in computer vision, the seminal work of Rusinkiewicz and Levoy [2001] on ICP algorithm comparisons led to significant progress in the scan registration field. The experiments employed simulated objects, highlighting different spatial constraints and sensor noises. Results mainly focused on the rapidity of convergence and the final precision of different solutions helping to select more appropriate solutions in further applications.

With the arrival of more standard sensors, researchers in robotics pushed the 2D registration algorithms so they can deal with larger environments with faster platforms [Bosse et al. 2004] and 3D slowly came back [Nüchter et al. 2004]. Because there exists no comparison framework, the selection of an appropriate variant for particular experimental conditions is difficult. This is a major challenge because registration is at the front-end of the mapping pipeline, and the arbitrary nature of its selection affects the results of all subsequent steps and more advanced robotic tasks. Even the early work of Eggert et al. [1998] highlights the difficulty to compare with other solutions given the lack of metric over common data sets. In their survey, Maintz and Viergever [1998] point the fact that proper accuracy studies are just starting; the problem being that the results provided are too specific. In addition, they highlight the imprecise use of the term accuracy, precision and robustness. They suggest to set up public databases and validation protocols, but foresee logistic, costs and efforts as incoming problem to those solutions.

Recently, the demand for a stronger experimental methodology in robotics was also stressed by Amigoni et al. [2009]. The authors survey different SLAM publications to highlight proper evaluation metrics that are applied to SLAM algorithms. Three principles of an experimental methodology in science (i.e, comparison, reproducibility/repeatability and justification/explanation) are translated in requirements for stronger SLAM results. As stated in their paper, a sound methodology should allow researchers to gain an insight about *intrinsic* (ex., computational time, parameters used, parameter behaviors) and *extrinsic* (ex., accuracy, precision) quantities. The authors report that, even though comparisons between algorithms are present in SLAM publications, very few researchers can reutilize the same protocol and directly compare their results without having to re-implement other solutions.

With the arrival of the Microsoft Kinect, another wave of publications is expected, as it was observed following the widespread utilization of SICK rangefinders. The Kinect augments accessibility to object modeling and indoor mapping research by having a handheld sensor connected via USB to a computer [Henry et al. 2012]. This camera produces depth reading as well as colors. This also opens the door to a mix of hybrid algorithms using labeled and unlabeled features without the need of expertise in sensor calibration.

1.4 Overview of the Thesis

Clearly, ICP is not a single solution, but more a framework where multiple variations and algorithms can be used to resolve geometric registration problems. For most of the applications, the pieces of the puzzle are spread in the literature but hard to assemble together. Therefore, in the light of this large corpus of work related to ICP and more generally to geometric registration, we address the following overarching research question:

How to select, evaluate, and develop appropriate 3D geometric registration solutions for autonomous mobile systems?

We believe that after 20 years of *new* registration algorithms, it is time to evaluate what works best for robotic systems. Therefore, our contributions aim at strengthening the current methodology and bring deeper analysis of current solutions. The timing is appropriate for such study given that computation power just reach real-time capabilities to support registration on embedded systems. Also, new electronic advancements brought more accurate and fast sensors in conjunction with better batteries allowing for longer operation time. Most importantly, researchers face a boom of solutions occluding a definition of standard solutions and avoiding the field to progress on algorithms that rely on registration like path planning, autonomous exploration, etc.

To answer our research question, we divided the work into three problems, for which we proposed solutions in the following chapters:

1. **The large number of solutions proposed**: a literature review of different solutions is presented in Chapter 2 with the aim to express ICP solutions in a common framework and validate our generic scheme proposed in Figure 1.3.

2. **The little amount of comparisons**: in Chapter 3, we propose a methodology to evaluate 3D geometric registration on common ground and we provide tools to support solution comparisons.

3. **The little amount of characterizations**: we provide a better understanding of current sensors limitations and present the results of our evaluation protocol applied to the most compared registration solutions in Chapter 4.

Finally, case studies using five different robotic platforms are described in Chapter 5. The requirements of each application are explained with some insight on how to tune parameters for specific applications. Those applications cover Search & Rescue activities, industrial inspection, shore monitoring and autonomous driving. All chapters close with a discussion in addition to a short summary. The conclusions of those chapter are recapitulated in Chapter 6 for final remarks.

Chapter 2
Formalization of ICP Solution Family

The spread of registration solutions renders tedious the task of finding particular problems to address, knowing there limits and applying them properly. Although well structured surveys exist to support registration in medical imagery [Maintz and Viergever 1998; Markelj et al. 2012; Fluck et al. 2011; Pluim et al. 2003] and object reconstructions/recognition [Salvi et al. 2007; Bowyer et al. 2006], there is no scientific publication covering the evolution of registration applied to robotics. Despite the fact that the effort of categorizing the work of others can be tedious, surveys structure the nomenclature and focus the attention on problem relevant to the field.

Different application fields have different constrains, which in our case motivate a more targeted survey for robotics. For example, medical imaging has the advantages of controlled environments, standardized sensors, precise sensor motions, etc. On the other side, it has its own challenges: deformable objects, low descriptor discrepancy, multi-modal sensing, high impact of failure on human life, etc. As for object reconstruction/recognition field, advantages can be: controlled to semi-controlled environments, human in the loop for sensor motions, low real-time requirements, contained volume of objects, etc., with challenges including loop closure, realistic representation of the model, etc. On the other side, robotic applications often has an unbounded object of interest: the scene. This scene can be considered as a rigid body with spurious uncontrolled dynamic elements. Moreover, the focus is more on the stability of the 3D reconstructions linked to the localization of a mobile agent. In exploration, this mobility implies a suit of sensors moved in an unknown environment by an autonomous system.

This can imply a long sequence of measurements obtained through time in a range of different environments. Overlaps between measurements can largely vary and density of the point cloud should not be assumed uniform.

This chapter attempts to build foundation for a survey specialized on the robotic field. Although this research mainly focus on 3D registration applied to robots, having a broader overview of existing data-association solutions would be relevant at this early stage. We follow the generic scheme presented in Figure 1.3 to classify current publications. Different types sources for `reading` and the `reference` are surveyed in Section 2.1. Then, transformation functions mainly used in the Euclidian space are covered in Section 2.2.2. Next, Section 2.3 presents different preprocessing functions generally used to better approximate the environment or to enhance the discrepancy between salient points. Furthermore, Section 2.4 lists different way to associate a `reading` with a `reference` and some optimization techniques. Section 2.5 describes ways to handle outliers generated during the matching process. In Section 2.6, we finally expose diverse way to minimize alignment error.

Given the number of publications covering this field, we present examples of works for every modules instead of the complete list. Nevertheless, the classification used should be generic enough to allow further publications to be added in the future. In the summary presented in Section 2.7, we list the classification terminology used in our registration overview and highlight some relations between them.

2.1 Reading and Reference Sources

Sources where the `reading` and the `reference` come from are highly related to the application requiring data association. Robotic applications are mostly using 2D laser rangefinder for indoor [Yoshitaka et al. 2006] and outdoor [Bosse and Zlot 2009a, 2008] localization. With the need for 3D localization, systems using rotating laser rangefinder recently get an increasing attention also for indoor [Armesto et al. 2010] and outdoor [Segal et al. 2009] environments. In computer vision, laser are used to reconstruct historical items in 3D [Pulli 1999; Druon et al. 2006] or simply 3D models of small objects [Liu 2010; Jost and Hugli 2002]. Laser rangefinders are also used in face recognition [Pan et al. 2010]. Geomatic applications use lasers for 3D aerial mapping [Kumari et al. 2009] and, more exotically, depth data can also come from Atomic Force Microscopy (AFM) [Jost and Hügli 2002].

Geomatic and aerospace field mostly refer to Light Detection And Ranging (LiDAR) instead of laser rangefinder.

The selection of a depth sensor depends on many different criteria. For example, when external power is not accessible, power consumption becomes critical. Also, a high maximum range of the sensor may become essential if an environment is large or difficult to access. Field of View (FoV) can also impact the design of mobile inspection tools since too narrow FoV will require extra actuation to offer a better coverage of the environment. We propose in Figure 2.1 an overview of the market by representing 25 sensors with respect to their maximal range versus their weight. Payload that can be carried by a mobile platform may be a limiting factor of small systems. This figure is intended to help in defining a first group of sensors that a platform could carry while quickly seeing the possible scanning range accessible to the platform. On the y-axis of the graph, we can divide the graph into airborne survey sensors with more than 1 km range, terrestrial survey sensors around 100 m to 1 km, and safety/robotics sensors under 100 m.

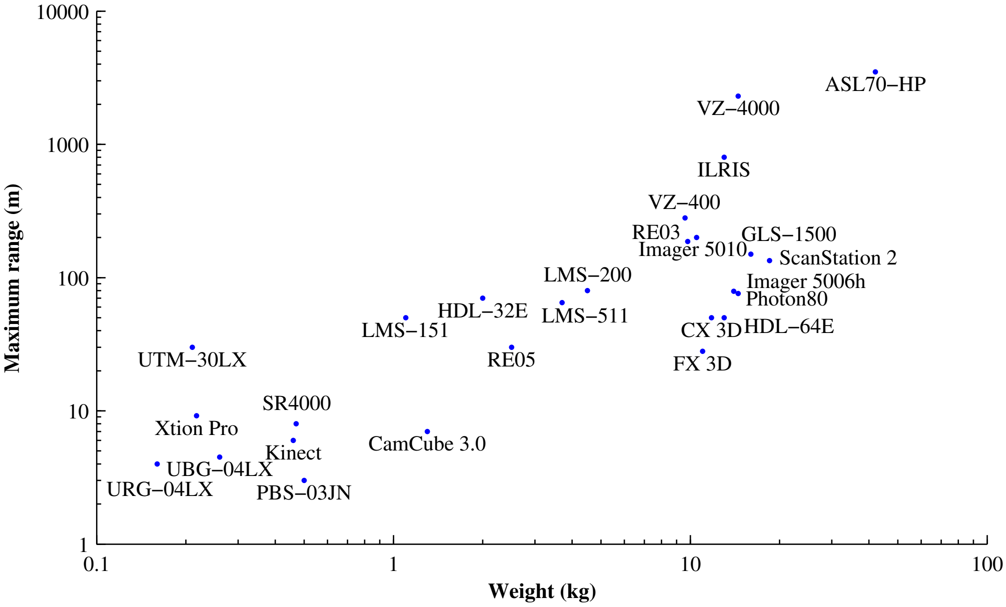


Figure 2.1: Classification of range sensors based on maximum range (m) and weight (kg). Note log scale. Manufacturers: Mesa Imaging (SR4000), PMD (CamCube 3.0), ASUS (Xtion Pro), Microsoft (Kinect), Occular Robotics (RE05, RE03), Hokuyo (UBG-04LX, URG-04LX, UTM-30LX), Sick (LMS-151, LMS-200, LMS-511), Velodyne (HDL-64E, HDL-32E), FARO (Photon80), Leica (ScanStation 2, ASL70-HP), Riegl (VZ-400, VZ-4000), Topcon (GLS-1500), Trimble (FX 3D, CX 3D), Optec (ILRIS), and Zoller+Froehlich (Imager 5010, Imager 5006h)

There are also new technologies emerging in this growing field. At the level of early products and development kits, it is worth noting the TigerEye 3D Flash LiDAR, proposed by Advanced Scientific Concepts, which can produce a depth image with a single laser beam. The system can read from a distance of up to 70 m with a 45° lens. Another company, Lytro, proposes a small and low cost camera producing depth images with a single lens. The system relied on a concept called *light field* to capture simultaneously multiple focus points and reconstruct images with different depth of fields out of the recorded data. Hand-held 3D scanner with sub-millimeter precision also exist for reverse engineering or culture heritage preservation, as offered by Artec 3D or Creaform to name a few. This type of sensors offers more precision at the expense of range.

Data association is also used in a variety of applications based on image acquired by a black and white or color camera [Zitová and Flusser 2003]. In medical applications, other specialized images, like red-free or fluorescein angiogram, are used to help diagnostic over several years of observations [Stewart et al. 2003; Tsai et al. 2010]. Two cameras can be registered together to produce a depth image which can then be used to track human body motions [Kim 2010] or create map of the environment [Diebel et al. 2004].

Zitová and Flusser [2003] propose a general classification of data acquisition as follow: different viewpoints (multi-view analysis), different times (multi-temporal analysis), different sensors (multimodal analysis), and also scene-to-model registration. Robotic and object reconstruction application are more related to the registration of different viewpoints while medical imagery different tend to do more time analysis to evaluate, for example, the status of a growing tumor. Calibration problem falls into the category of different sensors registration and search of a known object in the environment relate to scene-to-model registration.

2.2 Transformation Functions

Transformation functions allow to express an entity defined in a reference frame $\mathbb{A}$ in another reference frame $\mathbb{B}$. A first type of transformation functions without parameter can directly map spaces to another, following some common convention (e.g., mapping function from Cartesian to Spherical, Cartesian to Cylindrical, Homogeneous to Cartesian, etc.). A second type of transformation functions uses a set of parameters to move/deform an entity to express it in another frame.

2.2.1 Parametrized Transformation

Basic parametrized transformation functions are: translation, uniform scaling, rotation, nonuniform scaling, shear and projection. Typically, a combination of those basic functions is used. A short nomenclature can be introduced as follow:

- translation and rotation: *rigid transformation* (or euclidean transformation).
- *rigid transformation* and uniform scaling: *similarity transform*.
- *rigid transformation*, nonuniform scaling and shear: *affine transform*.
- vector and planar projection are from a group named *orthogonal projection*.

The projective transformation is not listed above because it is linked to a larger field, the projective geometric, which has ramifications beyond basic parametrized transformation functions described above. For further information, we refer to Appendix A, which expands more on reference frames and parametrization of transformation functions. Most of the registration algorithms used in robotics are based on *rigid transformation* parametrization. Nonlinear transformations are often expressed as a set of *rigid transformation* with some limited spacial influence. This type of registration can be referred as *flexible* registration and were considered first in medical imaging [Maurer et al. 1996; Feldmar and Ayache 1996] for organ reconstructions. It is interesting to note that those *flexible* registrations resemble to graph relaxation used for error back propagation in SLAM algorithm [Grisetti et al. 2007].

2.2.2 Initial Transformation

The `initial transformation` is a sensitive part of data association when the data association is realized mainly based on geometric features. We determined three types of `initial transformation` sources: (1) *unimodal*, where only one type of information is used; (2) *cascade*, where a geometric registration algorithm is used to feed the transformation to another one; and (3) *multimodal*, where multiple types of minimizations are used in sequences. Here are some examples of those initial transformation sources:

Unimodal While some papers present general algorithms assuming that a reasonable transformation will be given [Armesto et al. 2010; Druon et al. 2006; Schutz et al. 1998; Jost and Hügli 2002], others include humans in the loop [Godin et al. 1994; Pulli 1999]. More integrated solutions rely on external sensor (wheel encoders, Inertial Measurement Unit (IMU), Global Positioning System (GPS), etc.) as a first guest [Diebel et al. 2004; Yoshitaka et al. 2006]. Multiple initial guesses from the same external motion model and based on particle filters are also used in robotics [Grisetti et al. 2005] and in human body motion tracking [Kim 2010].

Cascade Cascade systems use the same registrations algorithm while vary parameters through the process to achieve faster or more accurate results. After some exit criterions, the registration parameters are changed and the outputted transformation is fed to the subsequent system. On example of cascade system is the coarse-to-fine strategy, where the `reading` or the `reference` are down sampled using different level of compression. The same registration technique is used to minimize the error sequentially from the most compressed layer to the less one [Zhang 1994; Jost and Hugli 2002; Magnusson et al. 2009; Bosse and Zlot 2009a]. A more rare example is the fine-to-coarse strategy, where the registration starts from a small bounding box. The expansion of the bounding box follows the uncertainty of the minimization until the bounding box covers the overlapping section of the `reading` with the `reference` [Stewart et al. 2003; Tsai et al. 2010].

Multimodal Multiple minimization techniques are used in sequence. The first registration is usually heavily based on descriptor matching because it is independent from the `initial transformation` (always identity). Then other types of registration, either using descriptors or features, are minimizing the alignment error. The motivation for the selection of those registration techniques is to reduce local minima possibilities with coarse alignment methods and continue with more precise but computationally more expensive methods. From the papers reviewed, system can have 2 layers [Bosse and Zlot 2008; Censi 2008; Godin et al. 1994; Stewart et al. 2003; Tsai et al. 2010] or 3 layers [Pan et al. 2010; Bosse and Zlot 2009a] but nothing seems to limit the number of layers implemented on a given system.

2.3 Data Filters

The different types of `data filters` try to augment the distinctiveness of the inputs usually by reducing the number of features and by augmenting the dimension of the descriptors. For example, a black and white image has a uniform distribution of features (a grid) and one dimension descriptor (the intensity) associated to each feature. After some `data filters` are applied, only few points in the image will be kept has features and the descriptors will be increased with information from neighboring pixels, typically to 64 dimensions (i.e., when using Scale Invariant Feature Transform (SIFT) descriptors [Lowe 2004]). In the case of a point cloud, it might be necessary to extract surface normal vectors (*feature enhancement*), while uniformizing the density of points (*features reduction*). This can also be viewed as *lossy data compression*.

In computer vision, remaining features are often called *key points.*

2.3.1 Feature Enhancement

When only geometric information is available, there are still ways to extract some level of distinctness by using Differential Geometry. We shortly introduce key concepts related to the use of geometric information. In this work, the notion of a shape S is used as a representation of an generic object in the Euclidian space with a certain set of properties. Those properties can be photometric, thermic, semantic, etc. Simple shapes, such as points, lines, quadrics, can be easily parametrized but most of the shapes encounter in the a real environment are too complex to be completely synthesized with parameters. To allow a certain representation of the world, a complex shape $\mathcal{S}$ can be approximated by a set of other shapes $\mathcal{P}$ only if they can be expressed in the same frame of reference $\mathbb{F}$.

$$^{\mathbb{F}}\mathcal{S} \approx {}^{\mathbb{F}}\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\} \tag{2.1}$$

Sensors measuring depth produce such approximation by discretizing the environment in a set of points. From smooth areas defined by those points, we present five types of primitives that can be extracted based on Differential Geometry: point, line, plane, curve and quadric. The first derivative group rely on normal vectors n (i.e. a vector perpendicular to a line or plane) and tangent vector t (i.e. a vector parallel to a line or plane) to express the area. Since they are defined with respect to a point p, normal and tangent vectors can be represented with a minimal set of 2 angles in polar coordinates. Normal and tangent vectors can be seen as a dual representation. The choice of using either one name or the other is defined by the minimum information required to express a primitive. In the case of a line in 3D, the normal vectors needs to define a plane perpendicular to that line. Therefore, only using the tangent vector is more convenient. The same reasoning hold for using a normal vector for the surface. The notion of direction also needs to be defined depending of the primitive. We choose to use unsigned direction in the case of tangents and signed direction in the case of normals where positive sign defined the outer surface of the shape. The motivation behind this choice is that we want to keep track of which side of a surface we are measuring while moving. In the 2D case, it is equivalent to represent a line by a tangent or a normal in term of the number of parameters. However, it is usually assumed that the perceived 2D plane cuts perpendicular surfaces, so it makes more sense to use normal vector to also track the outer side of the line. Figure 2.2 illustrates this choice of representation for the 2D case. Viewpoints v_n (i.e., where the sensor was when a point p was measured on a surface $\mathcal{S}$) are used to determine the direction of the normal vectors n, whereas no extra information is needed for a tangent vector t laying on a line.

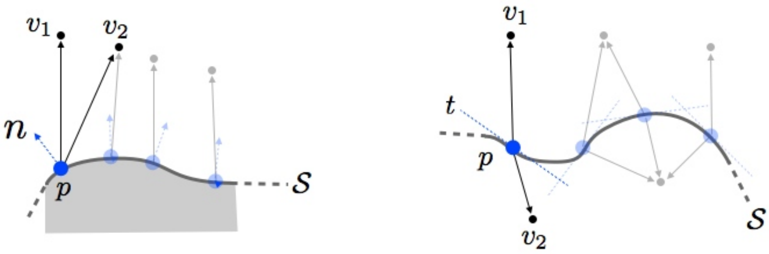


Figure 2.2: Difference between using normal vectors (left) and tangent vectors (right) to represent a 2D shape $\mathcal{S}$.

As for the second derivative group, a curve is parametrized by a curvature κ (a scalar) representing an osculating circle parallel to the tangent of a line. In the case of a quadric (i.e. a curved surface), it is parametrized by principal direction vectors t_{min}, t_{max} and the principal curvature scalars $\kappa_{min}, \kappa_{max}$. Again, the principal directions rely on a point p and on the normal vector n, so they could be expressed only with one angle. In other words, the principal directions are tangent vectors to the surface complementing the normal vector. In the case of curves, the curvature is always positive as opposed to quadrics for which a positive κ means that the surface bend in the same direction of the normal vector and vice-versa. Table 2.1 presents the minimum parametrization of those primitives. For simplicity in further computation, most of the publications use normalized vectors to express normals and tangents leading to a larger set of parameters, as resumed in Table 2.2.

Table 2.1: Minimum set of parametrization used to represent geometric primitives

Name	*Parameters*	*Constrains*	*Reference*
Point	$p = \{x, y, z\}$	$p \in \Re^3$	$\mathbb{F}$
Tangent	$t = \{\theta, \phi\}$	$\theta \in [-\pi, \pi[$ $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$	$\mathbb{F}$
Normal	$n = \{\theta, \phi\}$	$\theta \in [-\pi, \pi[$ $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$	$\mathbb{F}$
Principal Directions	ψ	$\psi \in [-\pi, \pi[$	$\{n, \mathbb{F}\}$
Curvature	κ	$\kappa \in \Re_+$	$\mathbb{F}$
Principal Curvatures	$K = \{\kappa_{min}, \kappa_{max}\}$	$K \in \Re^2$	$\{n, \mathbb{F}\}$

Table 2.2: Set of parameters usually found in the literature to represent geometric primitives.

Name	*Parameters*	*Constrains*	*Reference*
Point	$p = \{x, y, z\}$	$p \in \Re^3$	$\mathbb{F}$
Tangent	$t = \{t_x, t_y, t_z\}$	$\|t\| = 1$	$\mathbb{F}$
Normal	$n = \{n_x, n_y, n_z\}$	$\|n\| = 1$	$\mathbb{F}$
Principal Directions	$\gamma = \{t_{min}, t_{max}\}$	$n \perp t_{min} \perp t_{max}$	$\mathbb{F}$
Curvature	κ	$\kappa \in \Re_+$	$\mathbb{F}$
Principal Curvatures	$K = \{\kappa_{min}, \kappa_{max}\}$	$K \in \Re^2$	$\{n, \mathbb{F}\}$

Those parameters (point, tangent, normal, principal direction, curvature and principal curvature) brings us to a group of parametrized primitives (point, line, plane, curve and quadric), which can be helpful to approximate other complex 3D shapes. Those geometric primitives are listed in Table 2.3 with their characteristics. A graphical representation of those primitive is also showed in Figure 2.3 in the case of a shape considered as a 1D manifold and in Figure 2.4 in the case of a 2D manifold. In their current configuration, lines, planes, curves and quadrics are unbounded, which means that they can reach infinity on their unconstrained direction. One can constrained a primitive by using a primitive with a lower dimensionality [Besl 1988]. Then, a plane can be bounded by a set of lines, a line can be bounded by a set of points, etc.

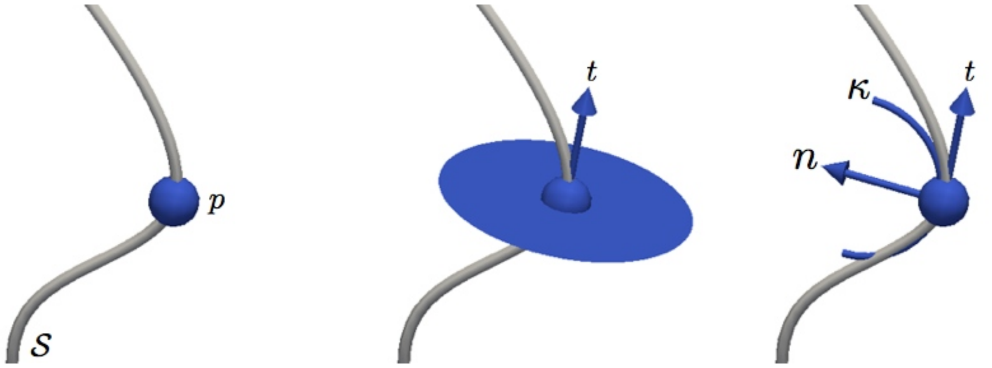


Figure 2.3: Representations (in blue) of a complex shape $\mathcal{S}$ (in gray) approximated as a 1D manifold. Left: no derivative (point). Middle: first derivative (line). Right: second derivative (curve).

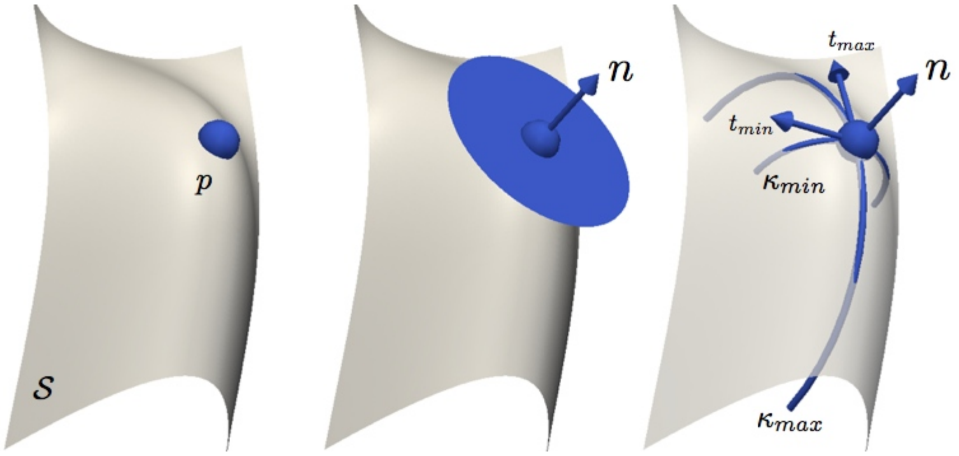


Figure 2.4: Representations (in blue) of a complex shape $\mathcal{S}$ (in gray) approximated as a 2D manifold. Left: no derivative (point). Middle: first derivative (plane). Right: second derivative (quadric).

Table 2.3: Characteristics of primitives used for shape approximations. Number in parenthesis of the column *Nb. Param.* correspond to the minimum number of parameters that can be used to express the same primitive.

Primitives	*Parameters*	*Derivative*	*Manifold*	*Bound*	*Nb Param.*
Point	p	0	0	-	3(3)
Line	$l = \{p, t\}$	1	1	point	6(5)
Plane	$\omega = \{p, n\}$	1	2	line, curve	6(5)
Curve	$c = \{p, t, n, \kappa\}$	2	1	point	10(7)
Quadric	$s = \{p, n, \gamma, K\}$	2	2	line, curve	14(8)

Higher derivatives could also approximate a shape at one point with more precision. Unfortunately, high derivatives are very sensible to noise when applied outside of a theoretical context. We limit ourself to the second derivative given that a non-negligible noise level is expected from the sensor measurements and from the motion of the sensor. We will see in Section 4.1.2 that even the first derivative needs a large supporting surface with typical laser rangefinders.

At an higher level of organization, a group of geometric primitives can be processed without any proximity assumption (*unstructured*) or with some smoothness constrains (*structured*). Examples of representation for a 1D manifold is a spline, while for a 2D manifold simple mesh or Non-uniform rational B-spline (NURBS) can be used. When it comes to a noisy group of points, tensor voting [Medioni et al. 2000] can be used to interpolate shape on a dense 3D grid. The voting results can then be later process to extract 1D and 2D manifolds out of the dense volume.

Sensitivity to transformation functions

The shape representations are affected differently by transformation functions. At a more generic level, transformation functions affect geometric quantities. Examples of quantities are: coordinate, orientation, length, angle, length ratio, etc. Those geometric quantities with examples of associated primitives are listed in Table 2.4. As example of lengths, we used κ, which is the inverse of a radius, and added the eigenvalues λ, which defined a scale over a vector. Having geometric parameters as much invariant as possible from transformation helps the matching function during registration because the association will be less sensible to large initial alignment error. Table 2.5 relates the different geometric quantities to the basic transformation functions affecting them.

Table 2.4: Quantities names susceptible to be affected by a transformation function with examples related to the parameters of geometric primitives.

Quantity	*Single Entity*	*Relationship in a Set*
Coordinate	p	-
Orientation	n, t_{min}, t_{max}	-
Length	κ, λ	$\|\|p_a - p_b\|\|$
Angle	-	$\mathrm{acos}(n_a \cdot n_b)$
Length Ratio	-	κ_a/κ_b

Table 2.5: Influence of a transformation function on quantities defining a geometric primitive. Cells marked with a "X" means that the transformation affect the values of the entity.

Function	*Coordinate*	*Length*	*Orientation*	*Angle*	*Length Ratio*
Translation	X	-	-	-	-
Uniforme Scaling	X	X	-	-	-
Rotation	X	-	X	-	-
Nonuniform Scaling	X	X	X	X	-
Shear	X	X	X	X	X
Orthogonal Projection	X	X	X	X	X
Perspective Projection	X	X	X	X	X

Most of the time, point cloud features come without external descriptor (ex.: intensity for image) so the proximity of other features is used to extend the shape approximation to further derivative. Surface orientations (or line orientations in 2D) are mainly used in literature [Pulli 1999; Censi 2008; Bosse and Zlot 2009a; Jost and Hugli 2002; Schutz et al. 1998; Jost and Hügli 2002; Segal et al. 2009]. Line orientations are also used in image registration where the environment presents very few salient points when considering only intensity variation [Stewart et al. 2003]. Surface normal vector distributions of surrounding points are also used by Magnusson et al. [2009] and Fairfield and Wettergreen [2009].

2.3.2 Descriptor Enhancement

A comparison of descriptors extracted from 2D point clouds can be found in [Bosse and Zlot 2009b]. It is proposed that moment grid is better than 2D shape context, Gestalt, Hough transform peaks, orientation and projection histograms, and normal orientation histogram grid. Extension to the 2D shape context can be found in [Tsai et al. 2010]. Another study for 3D point clouds also concludes that moment grids is better than 3D shape contexts, spin images, shell images and local covariance [Bosse and Zlot 2009a].

Usually, ICP is done using only geometric features but some works also present results using the laser reflexion intensity from an Hokuyo [Yoshitaka et al. 2006] and from specialized system using three different wave lengths [Godin et al. 1994]. Laser range finders are also combined with camera to add color information on measured points [Schutz et al. 1998; Druon et al. 2006]. When other sensors are used to provide descriptors, calibration of those sensors is required. Interestingly, calibration rely also on registration solutions. Terrestrial survey scanners often have a calibrated camera associating color to 3D points, just as the Kinect do. With the larger availability of photometric informations, the door opened to more classical solution from Computer Vision. In "A survey on local invariant features" [Tuytelaars and Mikolajczyk 2008], characteristics of descriptive features are listed as being *rotation, scale and affine invariant* and evaluation criterion as *repeatability, distinctiveness, locality, quantity, accuracy, efficiency*. In images registration, the list of most common tools extracting descriptors are: Harris, Hessian, SUSAN, Harris-Laplace, Hessian-Laplace, DoG (SIFT), SURF, Harris-Affine, Hessian-Affine, Salient Regions, Edge-based, MSER, Intensity-based and Superpixel [Tuytelaars and Mikolajczyk 2008]. It is interesting to note that descriptors based on photometric information count on passive illumination to ensure invariance. This rely on the assumption that the illumination point remain static which mostly true for indoor lights but one needs to be careful for outdoor illumination where the sun moves, clouds can shade light, etc. As for laser intensity measurements, they are even more sensible to transformation functions because the illumination point follow the sensor.

2.3.3 Feature Reduction

In application using point clouds, features arrive already sparse but not uniformly distributed. Nevertheless, the fact that sensors can provide a huge number of readings on a short period of time creates a bottleneck in term of computation power for the `match function`. Several techniques are used to reduce the number of features: random sampling [Jost and Hugli 2002; Pan et al. 2010], uniform grid [Magnusson et al. 2009; Bosse and Zlot 2009a], grid projection [Pan et al. 2010], octree [Fairfield and Wettergreen 2009; Wurm et al. 2010], and bounding box [Stewart et al. 2003; Tsai et al. 2010]. All these techniques reduce the number of features without considering their distinctiveness. Having that criteria in mind, Bosse and Zlot [2009b] present results showing that keeping a representative point per curvature clusters is better than segment centroids and mean-shift for 2D point clouds. Using color associated to each point, [Druon et al. 2006] uses 7 clusters based on hue values and selects only on cluster caring the most information for registration. Cluster segmentation is also used on laser intensity values [Godin et al. 1994].

It is also possible to find more application-specific methods in the literature. For example, the tip of the nose, inner eyes corner, and nose corners are directly extracted for face detection [Pan et al. 2010]. In medical imagery, blood vessel crossings are also used to reduce features. Moreover, the main orientation of the blood vessel crossings and its number of branching is used to construct descriptors [Stewart et al. 2003]. The complete point cloud can also be resumed to its first and second statistical moments [Liu 2010] or with orientation and projection histograms [Bosse and Zlot 2008].

2.3.4 Sensor Noise

Finally, sensor noise is also taken into account at this stage of the process. Those sensor noise models intend to evaluate the uncertainty of a measured point based on the limitations of the sensor used. They may try to identify if a point is a measurement artifact or how accurate is the position measured. To cope with stereo reconstruction noise, Diebel et al. [2004] removes points with distance and surface angle to neighbors larger than two times the median of all distances and surface angles within the point cloud. When using laser reflexion intensity, which is not invariant to distance and angle, Yoshitaka et al. [2006] proposes to keep points only close to the laser to avoid noisy measurements. For color images, points with low saturation value tend to be gray and are removed before applying clustering technique based on hue [Druon et al. 2006]. Points on boundaries of the sensor reading can also be removed to avoid misleading interpretation of neighbor points [Armesto et al. 2010]. When an error model is available, it is also possible to add noise information based on measurement distance, incidence angle, reflectivity, etc. Examples of noise models based on Gaussian is investigated in Section 4.1.

2.4 Match Functions

In the context of ICP, a `match function` $\mathcal{C}$ associates a first set of shapes $\mathcal{S}_{ref}$ to another set of shapes $\mathcal{S}_{read}$ based on a distance metric, and output tuples $\langle i, j \rangle$ representing links between shapes. Matching functions assumed that all shapes are represented in the same coordinate frame $\mathbb{F}$.

$$\langle i_{ref}, j_{read} \rangle = \mathcal{C}(\ {}^{\mathbb{F}}\mathcal{S}_{ref},\ {}^{\mathbb{F}}\mathcal{S}_{read})$$

2.4.1 Types of matching

The matching of the `reading` with the `reference` can be divided in three types: 1) features matching, 2) descriptors matching and 3) mixed. Feature matching is mainly done using Euclidian distance between a point in the `reference` and a point in the `reading` [Censi 2008; Druon et al. 2006; Pulli 1999; Pan et al. 2010; Kim 2010; Segal et al. 2009], a point and a plane [Champleboux et al. 1992] and quadrics [Feldmar and Ayache 1994]. Custom distances based on point positions and angles [Armesto et al. 2010] can also be used. Descriptors are matched based on their Euclidian distances [Lowe 2004; Bosse and Zlot 2009a]. The concept of measuring distance between two entities can take multiple forms (ex., correlation matching, earth mover distance, L^1, L^{inf}, etc.) In the current literature surveyed, other distances used in matching functions were Mahalanobis [Stewart et al. 2003] and Chi-2 test statistic [Tsai et al. 2010]. Both of those types of matching have their advantages and inconveniences. In laser rangefinder based matching, feature positions are quite accurate compared to descriptor uniqueness but the `initial transformation` needs to be within a maximum range to avoid local minima. When using descriptors, the matching becomes independent of the initial position but may fail for repetitive elements (ex.: checkerboard, building facades with repetitive windows, etc.).

When matching uses only descriptors, the `initial transformation` is not explicitly required. Although it is not mentioned, the transformation is treated as identity in those cases.

A logical extension is to mix both types of matching. One way is to express descriptors in the feature space using a conversion factor. This was used with surface orientations [Jost and Hügli 2002; Bosse and Zlot 2009a], surface orientations and color [Schutz et al. 1998], color [Johnson and Kang 1997] and laser intensity [Godin et al. 1994; Yoshitaka et al. 2006]. The other way around is also possible (i.e., expressing feature positions in descriptor space). In [Mortensen et al. 2005], a descriptor, called Global Context, is created using surrounding feature positions. The distance is computed as the sum of the Euclidian distance of a SIFT descriptor with the Chi-2 distance of the Global Context descriptor. A ratio between feature and descriptor distances can be used, but it is often implicitly defined as one [Tsai et al. 2010].

Other parameters to consider during the matching stage is the match direction and the number of matches used. The match direction refers to either match from the `reference` to the `reading` or from the `reading` to the `reference`. Most of the time, one of those two possibilities is used without further consideration but some techniques use both directions [Pulli 1999; Godin et al. 1994]. As for the number of matches, most of the applications consider only the closest point but some others process a certain percentage of the lowest distance [Stewart et al. 2003] or the complete matching matrix. The complete matching matrix is often used in loop closing detection [Bosse and Zlot 2009b] but it can be used for local matching, as with the SoftAssign method [Gold et al. 1998; Liu 2010].

2.4.2 Implementation optimization

The `match function` deals with the Nearest Neighbor (NN) problem which has a complexity of $O(nm)$ where n and m are respectively the number of elements in the `reading` and in the `reference`. This stage is generally the most time-consuming and a lot of papers present variations of NN search to reduce its complexity. A dynamic space partitioning can by applied using kD-trees to reduce the search complexity to $O(n \log m)$. Approximate kD-trees decreases the computational time by employing a distance thresholds to limit the search at the risk of returning sub-optimal neighbors [Arya and Mount 1993]. This increases the overall speed of the search, while the redundancy between points prevents large accuracy degradation [Nüchter et al. 2005]. In an iterative context, Nüchter et al. [2007] propose to use cached kD-tree for faster search. NN from the previous iteration are feed to the current search as starting points to accelerate the computation. Additionally, Zlot and Bosse [2009] compare kD-tree, locality-sensitive hashing and spill-trees and concluded that the kd-tree is better in terms of accuracy, query time, build time, and memory usage. They also observed that huge approximations can reduce the query time by two orders of magnitude while keeping a sufficient accuracy.

KD-trees provide very little acceleration for high dimensionality like the ones used for image based descriptors. Static space partitioning, usually based on grid, or hashing offer less adaptation but can compensate with their computation speed. Approximate search based on Best-Bin-First can be used instead for optimization [Lowe 2004]. Other techniques use dual proximity hypothesis (i.e., laser points ordered sequence [Censi 2008] and n-search [Jost and Hugli 2002]), projection on one grid [Pan et al. 2010] or on multiple grid (called multi Z-buffer) [Benjemaa and Schmitt 1997] to also reduce the search time. Although very useful, kD-tree also limits the distance used to be Euclidian. This forces some approximations when using Euclidian distances between components of unit surface normal vectors instead of the angle between those [Feldmar and Ayache 1996; Eggert et al. 1998; Gelfand et al. 2003]. Figure 2.5 illustrates the error between the angle distance of surface normal vectors against the approximation using Euclidian distance. One can notice that as long as the distance is low, the approximation can hold but with large error the discrepancy between the vector is lost. Moreover, the construction of the tree requires some time and often, only the `reference` is used as seeding points. Limiting the number of kD-tree constructions by the use of keyframe or metascans can help to decrease the registration time for a sequence of scans, while limiting the drift of the final path [Wulf et al. 2008].

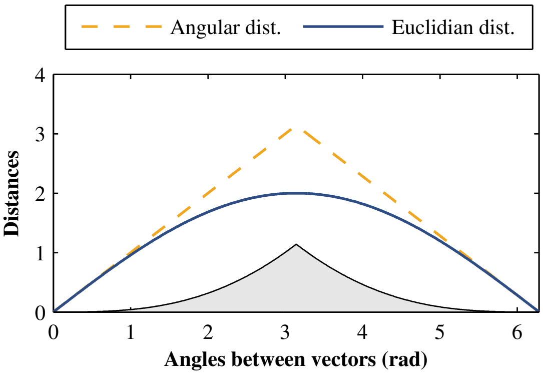


Figure 2.5: Impact of using Euclidian distance (dashed yellow line) instead of angle (solid blue line) between surface normal vectors. Shaded gray area represent the error.

Cascade systems, presented in Section 2.2.2, is also another research direction to accelerate the search. Jost and Hugli [2003] compute the ICP several times while varying the resolution from coarse to fine. At a coarse resolution (i.e., with a limited number of points) ICP converges faster but with less accuracy than at a fine resolution. However, by initializing a finer-resolution ICP with the result of the coarser one, the convergence of the fine-resolution ICP is much faster than with a single-shot ICP, as the initial alignment is mostly correct. These authors also used a pre-computed list of NN to approximate the matching step. With both of these techniques, they showed a significant increase of the speed of ICP while maintaining an adequate robustness. For the same absolute performance as standard ICP, Li et al. [2010] obtain less iterations at higher resolution, which decreases the total time by a factor of 1.5 in 2D and 2.5 in 3D. The multi-resolution approach can also increase the search speed for the closest point by using a hierarchical-model point selection with a stereo camera [Kim 2010]. By subsampling the space and with the help of the sensor structure, this solution can achieve a speed gain of factor 3 with respect to standard ICP when using a kD-tree search. In this case, the use of the structure of the depth image increases the matching speed. In the same direction, the specificity of a 2D laser scanner can help optimize the search [Censi 2008]. However, these optimizations are oriented toward specific sensors, which makes them hard to generalize, and are not suitable for a multi-sensor setup.

2.5 Outlier Filters

While the `data filters` try to reduce the impact of sensor noise, the `outlier filters` try to reduce the impact of wrong matches mainly caused by partial observations, dynamic elements and poor environment information extraction. Match quality can be evaluated based on feature pairs, descriptor pairs or even both together independently from the match distance used (i.e., based on features or on descriptors). When using descriptors, it is possible to apply an `outlier filter`. Regarding SIFT descriptors, the original method proposes to reject all matches where the distance ratio with the second match is higher than 80 % [Lowe 2004]. In the work of Stewart et al. [2003], all matches under 95 % confidence based on Chi-2 uncertainty bound are rejected.

In the case where the outliers are filtered based on features, rejection techniques are mostly using a threshold based on the Euclidian distance, the main difference being how to fix the threshold. A naive approach is to use a maximal distance between points [Segal et al. 2009]. This technique is sometimes hidden by using a fixed radius directly in the matching function. Likewise, surface orientation differences between paired points can be limited to a fixed value [Pulli 1999; Zhang 1994] or an adaptive one based on the median [Diebel et al. 2004]. Adaptive methods can be based on the mean distance between points and their Standard Deviation (STD) [Druon et al. 2006; Zhang 1994], STD only [Masuda et al. 1996], the quartile position (a.k.a trimmed) [Chetverikov et al. 2002; Censi 2008; Armesto et al. 2010] and the median [Diebel et al. 2004] of the distances of all paired points. In an iterative system for data association, [Pulli 1999] proposed to reduce manually the threshold at each iteration based on the notion of convergence of the system. An automatic extension to this approach is presented in [Pomerleau et al. 2010]. A different type of `outlier filter` evaluates whether there are multiple matches from the `reading` to the `reference` and keeps only the smallest distance [Zinsser et al. 2003].

2.5.1 Weightings

All the methods above are using *hard* assignment to identify outliers. Which means that, passed a certain threshold, the feature pair (i.e., tuple) is simply discarded prior to the minimization. Assignments can also be considered *soft* by using a weighting function promoting inliers during the minimization. Those weighting function can be custom such as the ratio of mean distance over each paired distances [Pan et al. 2010] or using known function such as Gaussian [Godin et al. 1994] and Cauchy (a.k.a. Lorentzian) [Bosse and Zlot 2009a]. Mix between *soft* and *hard* assignment are also used like one minus the ratio of the tuple distance divided by a maximum distance (with a saturation to zero when the tuple distance is over the maximum distance) [Diebel et al. 2004] or directly using the bisquare (a.k.a Tukey or Beaton-Tukey) function [Masuda 2001], [Stewart et al. 2003]. All those techniques use only the feature information to weight outliers but one paper computes a total weight by multiplying the feature distance by a reflectance similarity function [Godin et al. 1994].

2.5.2 Robust statistics

Dealing with outliers during a minimization process falls into the field of robust statistics. A suite of tools is proposed to robustly estimate the position (i.e., robust variant of the mean) and the scale (i.e., robust variant of the STD). For the scale estimation, some possibilities were found: Huber estimate, Median Absolute Deviation (MAD), interquartile range, Tukey estimator, trimmed estimator and Winsorised estimator.

The utilization of weights for a minimization process is based on a class of functions called M-estimators. Here is a list of M-estimators found in the literature: Least-squares (a.k.a L_2) (not robust and mostly used), Least-absolute (a.k.a L_1), $L_1 - L_2$, Least-power (a.k.a L_p), Fair, Huber, Geman-McClure, Logistic, Median, Talworth, Welsch, Cauchy (a.k.a Lorentzian). A category of M-estimator, called redescending M-estimators, have a saturation point to reject gross outliers. This category is equivalent to a mixed of soft and hard weights. Those functions are called: Hampel, bisquare and Andrews. Note that only the Cauchy [Bosse and Zlot 2009a] and the bisquare [Masuda 2001; Stewart et al. 2003] functions were found in data-association papers.

2.6 Error Minimization

The `error minimization` step relies on the definition of an error metric calculated from the association of features and needs to be resolved using an error model. The error model can be sometimes the same as the distance used at the matching stage but the main difference is that error is only defined in the feature space and not in the descriptor space. This is because only features are influenced by transformation parameters, as listed in Table 2.5. So, if the matching was based on descriptor distances, another error must be defined. Parameters selected for minimization should follow an expected deformation model. Zitová and Flusser [2003] present 2 generic types of error metric: global (rigid, affine transform, perspective projection model) and local (radial basis functions, elastic registration, fluid registration, diffusion-based, level sets, optical-flow-based registration).

2.6.1 Shape Morphing

Most of the data association algorithms based on point clouds use global-rigid error. This error metric is parametrized by 3 translations and 3 rotations parameters for a total of 6 Degrees of Freedom (DoF) when dealing with 3D point clouds (3 DoF in 2 dimensions). Point-to-point error uses the most basic primitive and was introduced in a registration context by Besl and McKay [1992] and used subsequently in multiple solutions [Godin et al. 1994; Pulli 1999; Druon et al. 2006; Pan et al. 2010; Kim 2010]. During the matching step, it might happen that different kind of geometric primitives (e.g., point, line, curve, plane, quadric) are matched together. Multiple error metrics were developed for those situations and we want to bring them under the same concept that we introduce as Shape Morphing. Essentially, when a primitive with higher dimensionality is matched with a lower one, it is morphed via projective geometry to adapt to it counter part. Figure 2.6 presents the list of possible combination for a 2D space and illustrates the concept for different errors.

The most represented example is the point-to-plane introduce by Chen and Medioni [1992] and then reused in multiple works [Champleboux et al. 1992; Gagnon et al. 1994; Bergevin et al. 1996; Gelfand et al. 2003]. Its 2D version, point-to-line, is also used in robotics [Bosse and Zlot 2009b] and a closed-form solution was presented by Censi [2008]. Using higher complexity to represent 3D primitives, Segal et al. [2009] proposes the use of plane-to-plane while early work of Feldmar and Ayache [1996] was already using quadric-to-quadric.

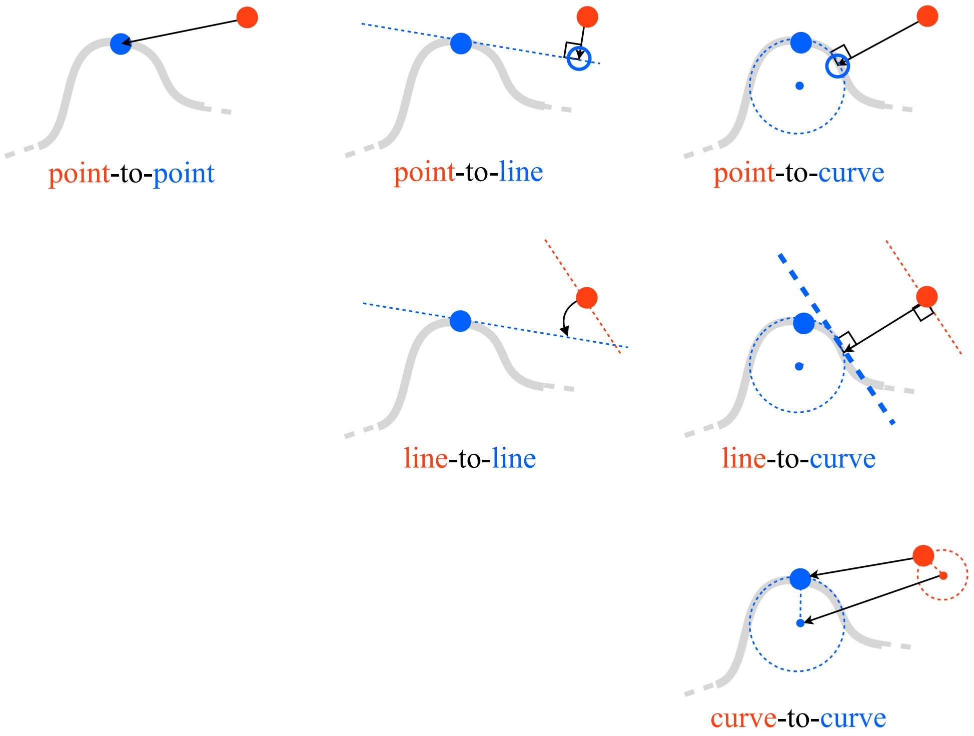


Figure 2.6: Possible morphing in 2D. The real shape is represented in gray with its approximation in blue. The resulting errors are represented with black arrows.

It is also possible to find extensions to those error metrics: point-to-point with extrapolation and damping [Zinsser et al. 2003], a mix of point-to-line with odometry error [Diebel et al. 2004], a mix of point-to-point, point-to-line or point-to-plane with angle [Armesto et al. 2010] and mix of point-to-point with Boltzmann-Gibbs-Shannon entropy and Burg entropies [Liu 2010]. Entropy based methods used in medical registration were reviewed by Pluim et al. [2003] as being: Shannon, Rodriguez and Loew, Jumarie, Rényi entropies. All those techniques rely on mean squared error.

Recently, [Silva et al. 2005] introduces a novel error called Surface Interpenetration Measure (SIM) which presents more robustness against different noise types. This measure was then applied later by [Pan et al. 2010] for face recognition. Image registrations mainly use affine transformations including skew and scale deformations like in [Lowe 2004]. A more complex hierarchical error models, presented by Stewart et al. [2003], increases the transformation parameter complexity from similarity to affine, reduced quadratic and finally quadratic. Those error models allow them to achieve higher precision on the final alignment while avoiding heavy computation at the beginning of the minimization.

2.6.2 Minimization

Once the error model is defined, a strategy or scheme to reduce the error must be selected. When a closed-form solution is available, direct minimization can be used. Unfortunately, closed solutions are rarely possible so two other minimization schemes are used: iteration and votes. Iterative schemes seem to be more applied to point cloud registrations. Within this category fall the well known ICP [Chen and Medioni 1991; Besl and McKay 1992], Normal Distributions Transform (NDT) [Biber and Straßer 2003], Simulated Annealing (SA) [Pan et al. 2010] and Genetic Algorithms (GA) [Silva et al. 2005]. Voting schemes are more the standard in image registration with Hough transform [Lowe 2004] and Random Sample Consensus (RANSAC) [Fischler and Bolles 1981]. Tensor voting was presented in a context of stereo image registrations [Medioni et al. 2000] and later applied to point cloud registrations [Reyes et al. 2007].

2.7 Summary

A survey of current registration solutions expressed in a common framework has been presented in this chapter. Publications on this topic have a large variety of contributions, from the adaptation of a generic solution to a specific application, up to a detailed theoretical solution of a single registration module. The concepts related to registration problem touch also a variety of mathematical tools (e.g., differential geometry, statistics, probabilities, (robust) regression, etc.) We also noted that same concepts have a different nomenclature depending if it is oriented for computer vision, robotics or medical imagery. Although the concepts are very similar, there is also a gap to be bridged between the terminology used for geometric registration or image registration. This situation seems to have been better handled in medical imagery than in robotics and computer vision. It could be explained by the fact that, from the beginning, registrations were applied for 2D/2D, 3D/3D and 3D/2D to cover the different sensors that medical staff were using to provide diagnostics. We provide a summary of key concepts in Table 2.6 to facility an overview of the classifications used.

Table 2.6: Possible classifications for registration algorithms.

`Reading` and `Reference`	
Sensor types	Photometric, time-of-flight, triangulation
Applications	Scene/object reconstruction, identification, tracking
Data acquisition	Different viewpoints, different times, different sensors, scene to model
`Initial Transformations`:	
Sources	External sensors, user, iteration
Types	Single hypothesis, multiple hypotheses
Systems	Unimodal, cascade, multimodal
`Data Filters`:	
Goals	Enhance discrepancy, reduce time, reduce noise
Descriptor invariance	Rotation, translation, scale, affine
Feature Relationship	Unstructured, structured
Support	Laser intensity, color, geometry
`Match Functions`:	
Types	Feature, Descriptor, Mixed
Direction	Reading to reference, reference to reading, both
Distance metric	Euclidian, Mahalanobis, Chi-2 test statistic, Custom
Optimization	Hashing/indexing, static space partitioning, dynamic space partitioning, feature reduction
`Outlier Filters`:	
Outlier sources	Partial observations, dynamic elements, sensor noises
Support	Features, descriptors, mixed
Assignment	Hard, soft, mixed
`Error Minimizations`	
Error	Geometric, morphing, entropy
Deformation	Global, local
Minimization schemes	Direct, iterative, voting

When a solution needs to be designed for a particular problem, requirements of the application should first be well clarified. Applications define `reading` and `reference` sources and how the `initial transformation` will be provided. As a rule of thumb, if the `initial transformation` can not be controlled to remain small, improvement on the `matching function` needs to be done, which may impact on the geometric representation built using `data filters`. `Outlier filters` are necessary to ensure a robust error minimization and, most importantly, to handle variable overlaps between the `reading` and the `reference`. One should compute only what is required for his application. Simple solutions often work well if the assumptions on which they rely are well understood and if the environment can be controlled to keep those assumptions true. We will demonstrate this statement using different applications in Chapter 5 based on system identifications realized in Chapter 4.

Chapter 3
Methodology

This chapter concentrates on the issue of scientific methodology raises by Amigoni et al. [2009] in the context of SLAM publications. We applied their observations more specifically to 3D registration problems. A complete scientific publication should provide not only a justification of the problematic and an explanation of the solution proposed, but also the means used to facilitate (1) comparison and (2) repeatability while demonstrating (3) reproducibility of the experimental work. A repeatable experiment needs to demonstrate sufficient measurements, while reproducibility refer to the ability of an experiment to be reproduced by an independent person. We address these three major points in the following sections while listing some guidelines for a better analysis of the results. Oriented more towards comparison and reproducibility, Section 3.1 presents two types of data sets specifically recorded for registration evaluations. The first data set group was introduced at the *AAAI 2011 Fall Symposium on Robot-Human Teamwork in Dynamic Adverse Environments* [Pomerleau et al. 2011a] and then published in the *International Journal of Robotics Research* [Pomerleau et al. 2012b]. Then, Section 3.2 introduces protocols facilitating a stronger repeatability while suggesting an evaluation list with the goal of accelerating comparison for further research. Section 3.3 focuses on reproducibility with an open source library specialized for registration. These two latter sections were introduced at the *IEEE 2011 International Conference on Intelligent Robots and Systems* [Pomerleau et al. 2011b] and then accepted for publication in the journal of *Autonomous Robots* [Pomerleau et al. 2013]. This chapter is concluded in Section 3.4 with a discussion on the online tools provided to accelerate research on geometric registration applied to robotics.

A notable example of an unreproducible experiment is the well covered story of the *cold fusion* claimed by Stanley Pons and Martin Fleischmann [Fleischmann and Pons 1989]. Since now, all further attempt to replicate the experiment failed.

3.1 Standard Data Sets

Many registration solutions have bloomed lately in the literature as demonstrated in Chapter 1 and Chapter 2. Although they are widely used, it is a common challenge to compare registration solutions on a fair base. The main limitation is the lack of accurate ground truth in current data sets [Smith et al. 2009], which usually cover environments only over a small range of organization levels [Pandey et al. 2011]. In computer vision, the Stanford 3D Scanning Repository pushes forward point cloud registration algorithms and object modeling fields by providing high-quality scanned objects with precise localization. We aimed at providing similar high-caliber working material for both the robotics and computer vision communities but with sceneries instead of objects. We proposed eight point cloud sequences acquired in locations covering the environment diversity that modern robots are susceptible to encounter, ranging from inside an apartment to a woodland area. The core of the data sets consists of 3D laser point clouds for which supporting data (Gravity, Magnetic North and GPS) are given at each pose. A special effort has been made to ensure a global positioning of the scanner within millimeter range precision, independently of environmental conditions.

3.1.1 Available Data Sets

Urban environment navigation has received quite some attention in the last years and triggered the creation of large-scale data sets of several kilometer long [Pandey et al. 2011; Huang et al. 2010; Smith et al. 2009]. Even though those data sets are undeniably very useful, other platforms, like the ones used for Search and Rescue missions, encounter a broader range of environments in which the robustness of localization needs to be assessed, which is difficult in with the available data sets. In Search and Rescue applications, environments that are likely to be faced are composed of complex structures, and some of them have particular problematic features such as a forest with dense foliage (see Figure 3.1) that shades GPS signals. On the registration side, the planarity of the environment was taken for granted in early implementations [Chen and Medioni 1991] and up to recent versions of scan matching algorithms [Pathak et al. 2010a]. Clearly, there is a need for semi-structured and unstructured data sets to challenge this planar hypothesis and to validate the robustness of registration solutions in a variety of environments that are encountered in the real world. Recently, Peynot et al. [2010] presented data sets that highlight various environmental situations, but the focus was on atmospheric conditions (airborne dust, smoke and rain). We continued in the same direction but for land-based studies by proposing data sets that cover a larger spectrum of environmental structures, so registration solutions can further be evaluated in real situations.

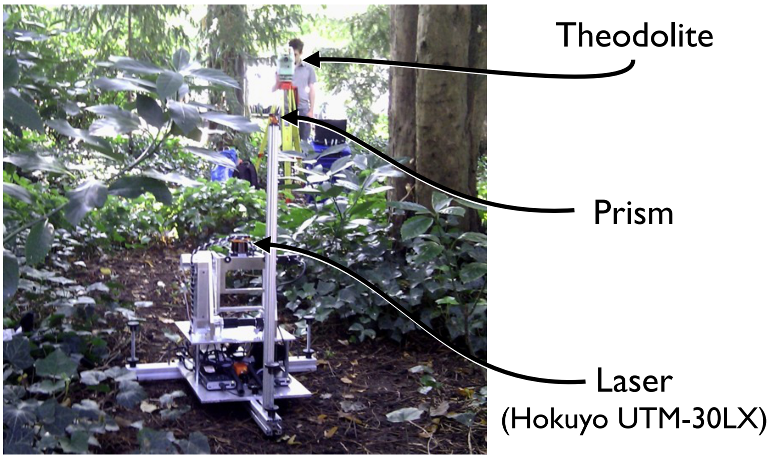


Figure 3.1: Scanner in targeted unstructured environments with dense foliage cover.

3.1.2 Ground Truth Localization

The notion of *ground truth* is highly dependent on the intention of use and can hardly be absolute. The error of the reference measurement used as ground truth should be significantly lower than the expected outcome of the algorithm to achieve a fair comparison.

Precise global positioning can be obtained using a mechanical arm that is fixed on a base holding a scanner, but this solution offers a limited motion range. On the other side, GPS and differential GPS systems can accommodate a large range of motions but are limited to outdoor locations displaying a clear sky condition. The precision of such systems can be highly variable (i.e., depending of foliage coverage, satellite alignment and number, multi-paths, etc.), which also limits the evaluation of registration precision. Optical motion capture systems, like the one proposed by Vicon, have recently appeared as a precise way to track sensor poses [Pomerleau et al. 2011b]. Those systems offer millimeter precision at 100 Hz, but could hardly be installed outdoors or in a highly cluttered environment. Instead of using fixed sensors and mobile markers, Tong and Barfoot [2011] propose a methodology to reuse directly laser reflectivity readings combined with some reflective beacons. This is a convenient way to acquire ground truth localization in open space, but would lead to the installation of multiple landmarks in a highly occluded environment, like a forest. Finally, Jet Propulsion Laboratory uses a theodolite to track specialized prisms fixed on a mobile platform to validate visual odometry performances [Maimone et al. 2007]. The precision reported was less than 2 mm in position, and less than 0.2° in attitude. In addition to the excellent precision, the system reduces infrastructure installation and ensures a fixed precision over all recorded sequences, independently of environmental locations and conditions, which is why we applied this technique to our data sets.

For our data sets, all sequences were selected to evaluate point cloud registration algorithms with respect to: (1) semi-structured and unstructured environments, (2) rapid variation of scanning volumes, (3) repetitive elements, and finally, (4) dynamic elements. Given that we targeted global positioning evaluations, a special attention was given to the methodology used to record ground truth poses with a consistent protocol for all the sequences.

3.1.3 Material

The data sets were recorded with a partially custom-made rotating scanner used in conjunction with a theodolite, as depicted in Figure 3.2. The main sensor of the scanner was a laser rangefinder (Hokuyo UTM-30LX) mounted on a tilting device. The sensor has a compact size (87×60×60 mm) and covers a field of view of 270° with a reading at every 0.25°. The precise control of the motor was ensured by a Maxon Motor EPOS controller. The control system put in place used a dual regulation loop based on two encoders. One encoder was located directly on the motor shaft to provide stability control while the second was located at the end of the transmission chain. Encoders had respectively 2000 and 48000 ticks per revolution, the precision difference coping with the gear reduction employed. The later encoder gave us a resolution of 0.0075° on the tilting axis. This setup allowed to remove the uncertainty from gear backlash and transmission strap deformation, which was estimated around 5°. Supporting data (Gravity, Magnetic North and GPS) was provided by a consumer grade GPS-aided IMU, Xsens MTi-G.

The theodolite utilized was the Total Station (TS15) from Leica Geosystems. Because it only measures one position at a time, three measurements are necessary to retrieve the complete pose (translation and orientation). A specialized reflective prism was mounted on a pole, which could be secured at

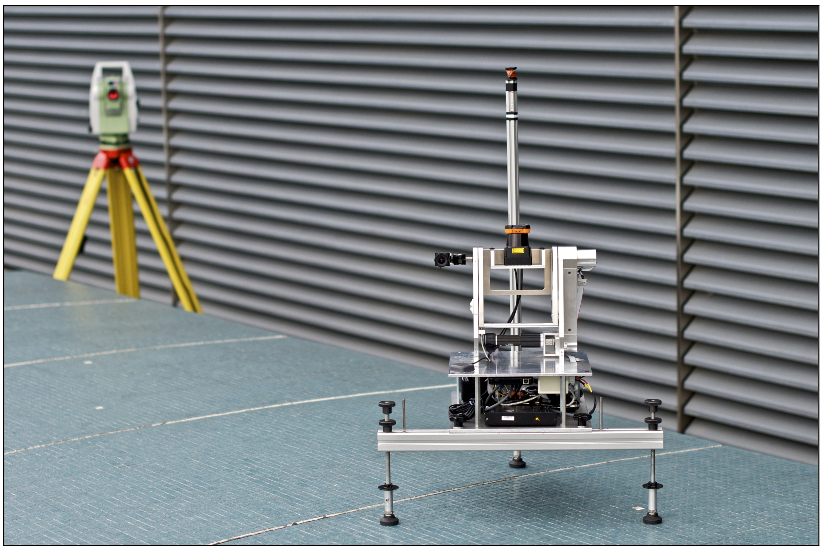


Figure 3.2: Tilting scanner with the prism mounted at p_0. The theodolite used for ground truth measurements is in the background.

three different locations on the scanner, namely p_0, p_1 and p_2 (see Figure 3.3 (a)). A steel guide ensured the pole to be positioned at the same location on the scanner every time. The pole was higher than the scanner to reduce visual occlusion from the theodolite.

3.1.4 Field Deployments

Most of the recording process was done manually. The scanner was moved from a location to another by an operator. Extra precautions were taken to ensure that the scanner stayed in place while scanning (usually for 20 s) and while the ground truth pose was measured (less than two minutes). Rubber feet were used on hard floors, and metal spikes were used on soft grounds. The inertia of the platform also guaranteed a good stability while recording a scan. In some cases, like a in compartmented area such as an apartment, a single line of sight cannot track all poses. For those situations, we changed the theodolite pose and then used the last scanner pose as a fixed beacon to relocalize globally the theodolite. We carefully planned those re-localizations to minimize their numbers so that for all the sequences, we never had to relocalize more than twice. We acknowledge that the overall system is costly and time consuming (e.g., three hours for 35 scans), but we firmly believe that this methodology is necessary to ensure that high-quality data sets are available for further research.

All sensor data were logged on the same computer so the data were time-stamped based on the same clock. Supporting data were recorded at a

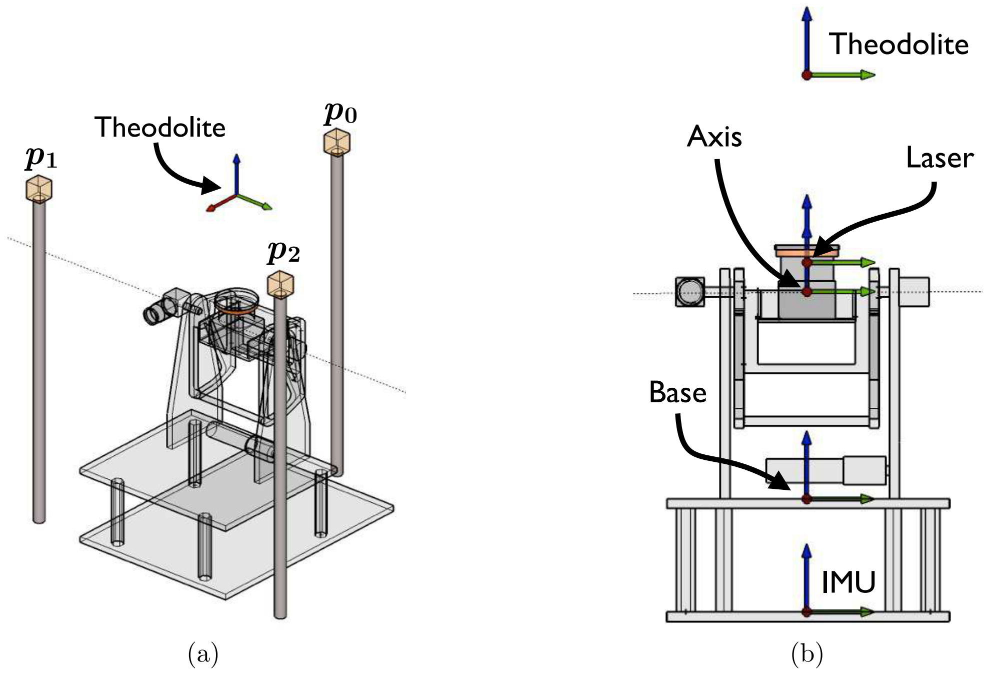


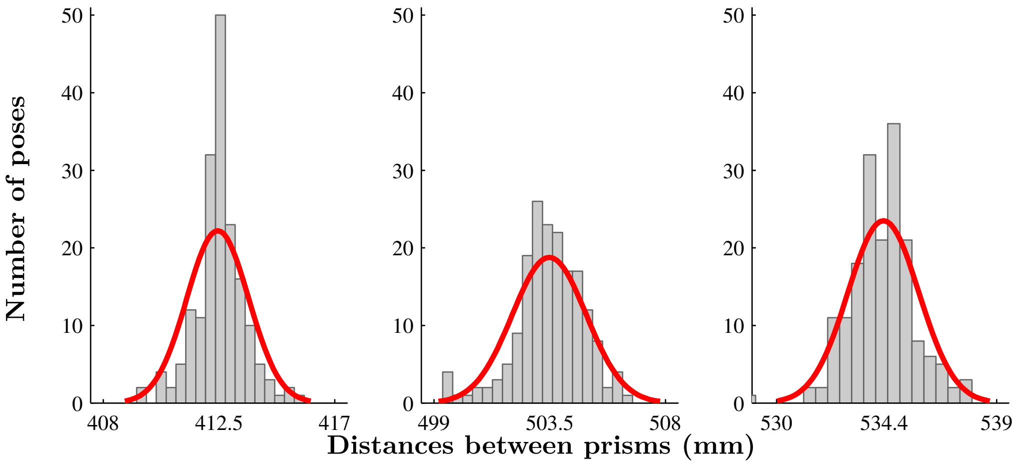
Figure 3.3: Configuration of the scanner. The dashed line corresponds to the rotation axis. (a) Perspective view with positions of the 3 prisms used to reconstruct the global pose. (b) Reference frame notation.

different frequency than the laser, and they were segmented per 3D scan pose.

3.1.5 Noise Evaluation

This section overviews the sources of noise, from the global pose down to a measured laser point. As claimed by the manufacturer, the theodolite has an accuracy of 1 mm per kilometer. Because we did not have access to an additional and more precise sensor to validate the ground truth, we evaluated the distances between each prism (d_{12}, d_{02} and d_{01}) over 181 scanner poses measured in different conditions (Figure 3.4). The maximum standard deviation (σ_{max}) of the three distances is 1.4 mm. Since we plotted distances between two prisms positions, we can assume that the STD of one point is approximatively 1.0 mm. This error englobes the noise of the theodolite and some operator manipulation errors while moving the prism from one position to another.

In the field, we used those position statistics to cancel spurious pose measurements before taking the 3D scan. The translation component of



(a) $\mu_{12} = 412.5\,\text{mm}$ $\sigma_{12} = 1.2\,\text{mm}$

(b) $\mu_{02} = 503.5\,\text{mm}$ $\sigma_{02} = 1.4\,\text{mm}$

(c) $\mu_{01} = 534.4\,\text{mm}$ $\sigma_{01} = 1.4\,\text{mm}$

Figure 3.4: Distribution frequencies of the distances between prisms [mm] measured by the theodolite. (a) Distances between p_1 and p_2. (b) Distances between p_0 and p_2. (c) Distances between p_0 and p_1. The mean μ and the STD σ of the theoretical Gaussian curves are plotted over the histogram of the distances.

the global pose is obtained with the mean of the three prism positions, which would again lead to a σ_t of $\frac{1.0}{\sqrt{3}} = 0.58\,\text{mm}$ under the assumption of isotropic Gaussian noise. For the rotational components, we used the smallest estimated distance between prisms ($\mu_{12} = 412.5\,\text{mm}$) and when using basic geometry from Figure 3.5, we can estimate an angular error (σ_θ) of 0.003 rad. Those errors applied if the theodolite is kept at the same place during a complete data set recording, which was not the case for three sequences (see Table 3.1). After a simple error propagation, we can approximate the global position error σ_t to be under 1.8 mm and the orientation error σ_θ under 0.006 rad, which is consistent with the level of precision reported by Maimone et al. [2007]. As for the transformation linking the *Theodolite* to the *Base* (Figure 3.3) frame, the parameters were computed with a global optimization technique based on registration results in a controlled and highly structured environment. To evaluate the alignment errors, we used as control a different data set then the one used for the optimization.

As to the transformations from the *Laser* to the *Base* (Figure 3.3), most of them were taken from the construction plans and were machined with a precision under millimeter in centimeter thick aluminum plates. Since encoders work in relative position, a homing procedure needs to be

Sequence Name	Nb. Scans	Nb. Pts / Scan	Re-loc.	Poses Volume (x× y × z)	Scene Volume (x× y × z)
ETH Hauptgebaude	36	191 000	0	24× 2 ×0.50 m	62× 65 × 18 m
Apartment	45	365 000	2	5 × 5 ×0.06 m	17× 10 × 3 m
Stairs	31	191 000	0	10× 3 ×2.50 m	21×111× 27 m
Gazebo Summer	32	170 000	1	5 × 4 ×0.07 m	35× 45 × 16 m
Gazebo Winter	32	153 000	1	4 × 5 ×0.09 m	72× 70 × 19 m
Mountain Plain	31	102 000	0	18× 6 ×2.70 m	36× 40 × 8 m
Wood Summer	37	182 000	0	10×15×0.50 m	30× 53 × 20 m
Wood Autumn	32	178 000	0	6 ×12×0.50 m	36× 60 × 22 m

Table 3.1: Characteristics of the point clouds for each sequence.

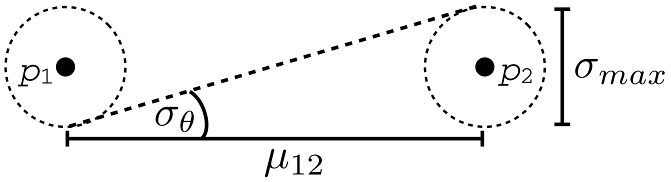


Figure 3.5: Worst case orientation error given the position error σ_{max} and the smallest expected distance μ_{12} between the prisms p_1 and p_2

applied to reset the count of the encoders. The offset between the homing position and the position of the rotating frame that is parallel to the *Base* is directly added in the low level controller (Maxon EPOS). This offset has been measured using two laser pointers, the ones typically used for public presentations, fixed on the tilting *Axis* and on the *Base*. The two laser points were projected on a wall at a distance of 8 m. The angle was adjusted to ensure that the distance between the projected laser points and the laser pointers were the same. We roughly estimated the homing error σ_h to be under 0.001 rad.

Finally, the Hokuyo UTM-30LX is a time-of-flight sensor with a minimum range of 0.1 m and a maximal range of 30 m. The specifications of the sensor indicates an accuracy σ_r varying from 0.01 m to 0.03 m depending on the distance and reflectivity of the targeted object.

Values for transformations from the different frames depicted in Figure 3.3 (b) are listed in Table 3.2 with their estimated precision. We used a right-handed coordinate system with the x-axis pointing forward, y-axis on the left and z-axis upward. All the transformations were given the notation $T_{X \leftarrow Y}$, which can be read as: a transformation T that can express a point, originally in the Y coordinate frame, in a X coordinate frame. Translation vector t is represented as $[t_x, t_y, t_z]$ and the rotation vector q is represented as a quaternion $[q_x, q_y, q_z, q_w]$, where q_w is the real part of the quaternion.

	Sensor	Estimated Pose	Estimated Precision
$T_{T\leftarrow G}$	*Global* to	t = variable	$0.0006 < \sigma_t < 0.0018$ m
	Theodolite	q = variable	$0.0030 < \sigma_\theta < 0.0060$ rad
$T_{B\leftarrow T}$	*Theodolite* to	t = [0.016 -0.024 0.606] m	residual = 0.004 m
	scanner *Base*	q = [0.000, 0.010, -0.006, -0.999]	residual = 0.004 rad
$T_{B\leftarrow A}$	Tilting *Axis* to	t = [0.000, 0.000, 0.220] m	by construction
	scanner *Base*	q = variable	$\sigma_h < 0.001$ rad
$T_{B\leftarrow I}$	IMU to	t = [0.000, 0.000, -0.085] m	by construction
	scanner *Base*	q = [0.000, 0.000, 0.000, 1.000]	by construction
$T_{A\leftarrow L}$	*Laser* to	t = [0.000, 0.000, 0.040] m	by construction
	tilting *Axis*	q = [0.001, 0.000, -0.003, 0.999]	by construction
$T_{L\leftarrow p}$	*Point* to	$r \in$ [0.1, 10) m	$\sigma_r < 0.01$ m
	Laser	$r \in$ [10, 30] m	$\sigma_r < 0.03$ m

Table 3.2: Relative transformation between frames with their estimated precisions.

As a general observation, very small angular misalignments can have a large impact on point location at large distances, especially for highly slanted surfaces. For example, we had to tune manually the orientation of *Laser* to the tilting *Axis* by a third of a degree to ensure that a single point cloud joints properly after a rotation of 180°. This slight offset might be due to tolerances in the construction and is related to the divergence of the laser beam that is typically around 1°. Although the precision of the scanner global pose is in the order of millimeter, it is most likely that the uncertainty of the reflected points in the environment is way larger when the beams have a diameter of several centimeters at a few meters of distance. This uncertainty is inherent to the sensors and occurs in most robotic systems.

3.1.6 Overview of the Data Sets

The aim of the proposed data sets was to provide unregistered point clouds for researchers who are seeking to evaluate their registration solutions on a common base. The point clouds were provided in *Base* frame, because it can be compared against the measured global poses. We also provided globally consistent point clouds for researchers doing environmental modeling. Before presenting the specific sequences, we first introduce the nomenclature employed to characterize the different sequences. The abbreviations defined below are reused in Table 3.3, which also presents an overview of the eight sequences recorded. The organization of the environment is characterized as follows:

Structured (S): The environment can mainly be explained by geometric primitives (e.g., offices or buildings).
Unstructured (US): The environment mainly involves more complex structures (e.g., a dense forest or a very untidy room).
Semi-structured (SS): The environment has both geometric and complex elements (e.g., partially collapsed building or a park essentially composed of a flat ground and some trees).

Considering a static sensor pose, we also defined three types of dynamic elements:

Intra-scan motions (AM): An element is moving while the data are captured. The longer time it takes to capture the data, the more deformed the element will be (e.g., walking persons or cars). This is comparable to motion blur for a fixed camera.
Inter-scan motions (EM): A dynamic event occurs punctually with respect to data acquisition (e.g., moved furniture or doors opened).
Global motions (GM): An event affects the environment at a global scale, and dynamic elements are detected by multiple views recorded at different time periods (e.g., seasonal changes or a building collapsing).

Finally, environment locations are divided into two categories: **Outdoors** (OUT) and **Indoors** (IN).

The sequences were recorded over half a year (between August 2011 and January 2012). Figure 3.6 and Figure 3.8 present a visual overview of all sequences showing the variety of environments covered. Table 3.1 gives the number of 3D scans, the average number of points per 3D scan and the number of times the theodolite was relocated for each data set. The two last columns give an indication of the volumes covered with a bounding box in which the scanner was moved (*Pose Volume*) and with a bounding box of the global map (*Scene Volume*).

Sequence Name	IN	OUT	S	SS	US	AM	EM	GM	Particularities
ETH Hauptgebaude	✓		✓			✓			Repetitive elements like pillars.
Apartment	✓		✓				✓		Single floor apartment with 5 rooms.
Stairs	✓	✓	✓						Rapid variations of scanning volumes.
Gazebo (x2)		✓		✓		✓	✓	✓	Recorded in summer and in winter.
Mountain Plain		✓			✓				Pasture with few vertical structures.
Wood (x2)		✓			✓	✓		✓	Recorded in summer and in autumn.

Table 3.3: Overview of the sequences with their characteristics.

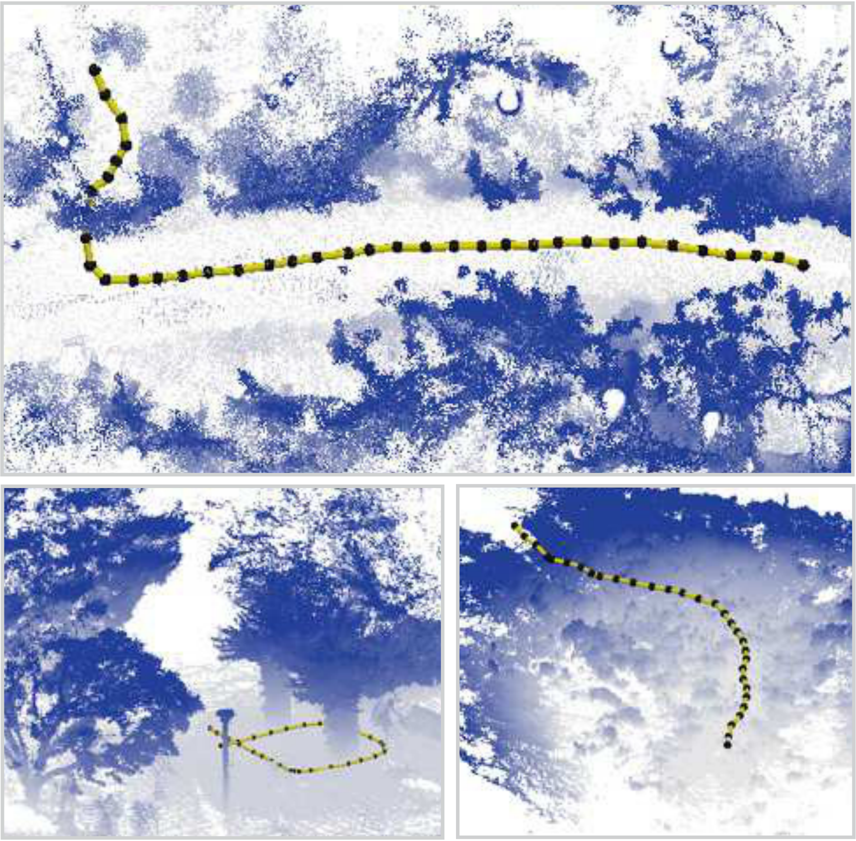


Figure 3.6: Unstructured and semi-structured data sets. Top: aerial view of *Wood* with the upper part of the vegetation removed. Bottom left: part of *Gazebo* with the wine trees on the right and some large trees on the left. Bottom right: aerial view of *Plain*. For all figures, the lines and black spheres correspond to the scanner poses and point clouds were colored to emphasize the depth of the structure from the virtual camera perspective.

3.1.7 Unstructured and Semi-Structured Environments

The sequence named *Wood* was a good example of a challenging environment for registration algorithms that contains both complex structures and intra-scan dynamic elements. Figure 3.1 shows the starting position of the recorded path. This environment was mainly constituted of vegetation (trees, bushes, etc.), a small paved road crossing the wood being the only structural element. While recording the data, some people were walking on the road. The scanner path started in the wood and continued for approximately 12 scans before joining the small road for the next 14 scans. The sequence was recorded at two different seasons (i.e., summer and late autumn), which gives the opportunity to test registration algorithms robustness against Global Motion (i.e, seasonal changes). Figure 3.7 shows a visible example of the impact of those changes on trees, which trees were manually extracted from the global map for each season.

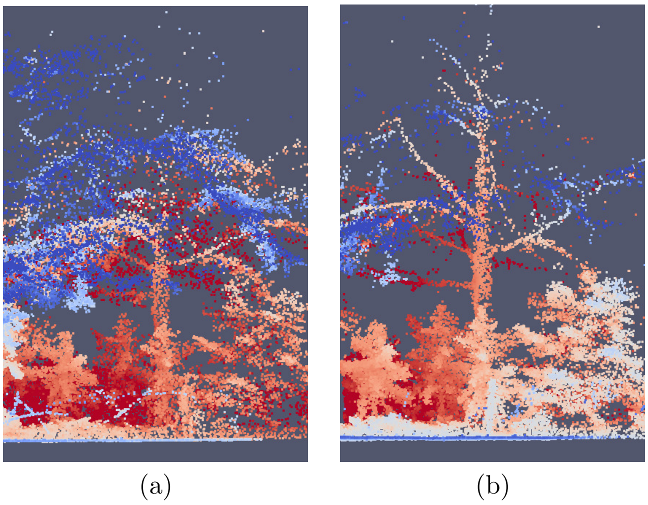


Figure 3.7: Extracts of global representations highlighting seasonal changes: (a) summer, (b) late autumn. Point cloud colors were selected to enhance depth of the screenshot.

Another sequence called *Mountain Plain* was recorded on a small area of an alpine plain located at 1920 m altitude. There was no major vertical structure in the environment and the main element on the ground is dry vegetation (around 50 cm height). The motivation behind this sequence was to evaluate robustness of registration algorithms against low-constrained, unstructured environment. The opposite of a low-constrained environment would be an apartment where the ceiling and walls are large enough to fix the position and orientation of the sensing platform easily. This sequence was also very interesting because the hypothesis of a planar motion of the scanner does not hold since the scanner is going down a hill before ending in a flat area.

The two sequences named *Gazebo* were recorded in summer and winter in a park, in which there were grass, paved small roads and sparse trees. The main construction standing in the environment was a gazebo with rock walls and a ceiling covered with wines trees. This place is a good representative of semi-structured environments with a mixture of man-made constructions and vegetation. Some people were walking while the scanner was recording, whereas others stayed seated for several scans under the gazebo. Figure 3.6 shows virtual views of those sequences with the path realized by scanner during the recording.

3.1.8 Other Environments

To ease comparisons between a more complete spectrum of environmental structures, we also provided three more sequences recorded with the same methodology. The sequence called *Stairs* aimed at evaluating robustness of registration algorithms against rapid variations of scanned volumes. The path started indoors, crossed some doorways and finished outdoors. The scanner passed over five steps, which offered a more complex motion than a flat floor. *ETH Hauptgebaude* tackled the problem of repetitive elements with its multiple pillars and arches in the hallway. In registration minimization context, those elements may create multiple local minima, which can trigger interesting observations for robustness evaluations. Finally, the sequence *Apartment* was a well-structured environment including: a kitchen, a living room, a bathroom, an office and a bedroom. A special care was taken to include outer-scan motion by moving a person, some furniture and boxes in between scans. The registration complexity of this environment was considered low, so it could be used as a reference for other types of environments. In both sequences *ETH Hauptgebaude* and *Apartment*, the scanner moved indoors on a flat ground.

3.1.9 Contextual information

In order to assist the researchers as much as possible in their experiments, we provided a certain number of contextual information for each sequence. We evaluated the overlap between all pairs of scans and provided the information as overlap matrices. Figure 3.9 shows those overlap matrices with the index of the scans (temporally growing) used on both axes. First, one can see that the overlap is not exactly symmetric. Indeed, if a scan cover a smaller area

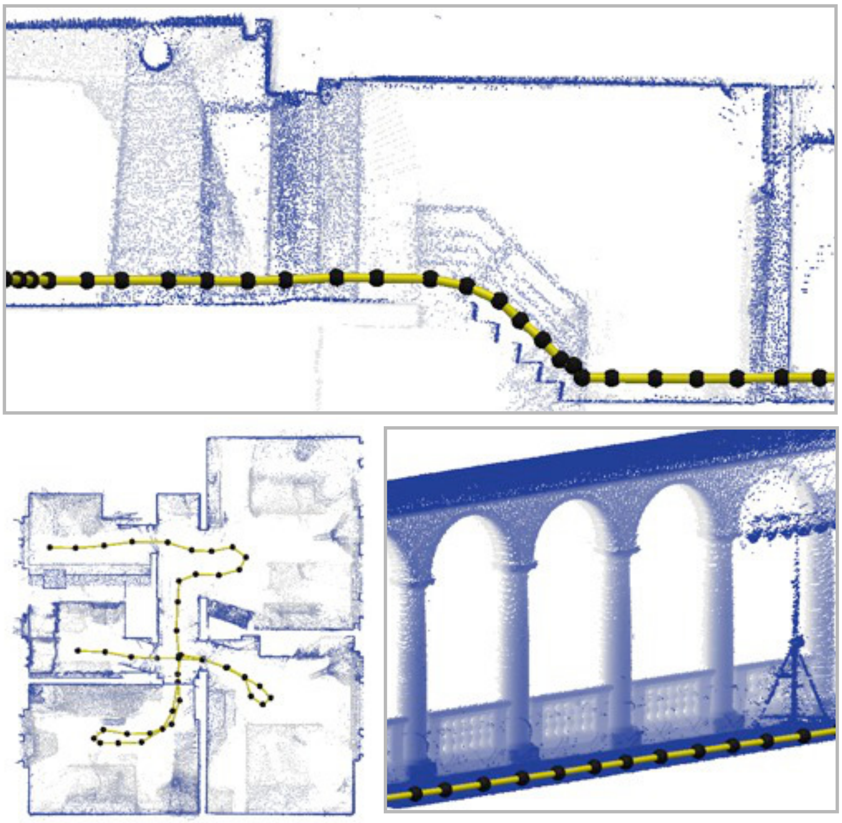


Figure 3.8: Structured data sets. Top: side view of the *Stairs*. Bottom left: top view of the *Apartment* with the ceiling and floor removed. Bottom right: cut view of a hallway from *ETH Hauptgebaude* showing arches and pillars. For all figures, the lines and black spheres correspond to the scanner poses and point clouds were colored to emphasize the depth of the structure from the virtual camera perspective.

than the other, all its points will find a match in the second, but not the other way around. Second, *Apartment* and *Stairs* show clusters of scans with high overlap within themselves but low overlap with other clusters of the same sequence. This is due to the segmentation of the volumes in the environment; typically, scans inside a room will all have a relatively high overlap while in between rooms the overlap will quickly drop. In comparison, *ETH*, *Wood*, and *Mountain Plain* share a pattern showing a high overlap that decreases as the index difference grows, as expected. Finally, *Gazebo* shows relatively high values of overlap for each of its scans because the environment is rather open, and there are few occlusions. We would expect *Mountain Plain* to show also high overlap ratio but it is not the case due to the ground configuration, which is quite uneven, and the lack of points upwards and sideways, which can be confirmed by the average number of 3D points per scan as shown in Table 3.1.

Moreover, we also provided 2D topological representations with annotations for the different theodolite poses needed for the recording and extra

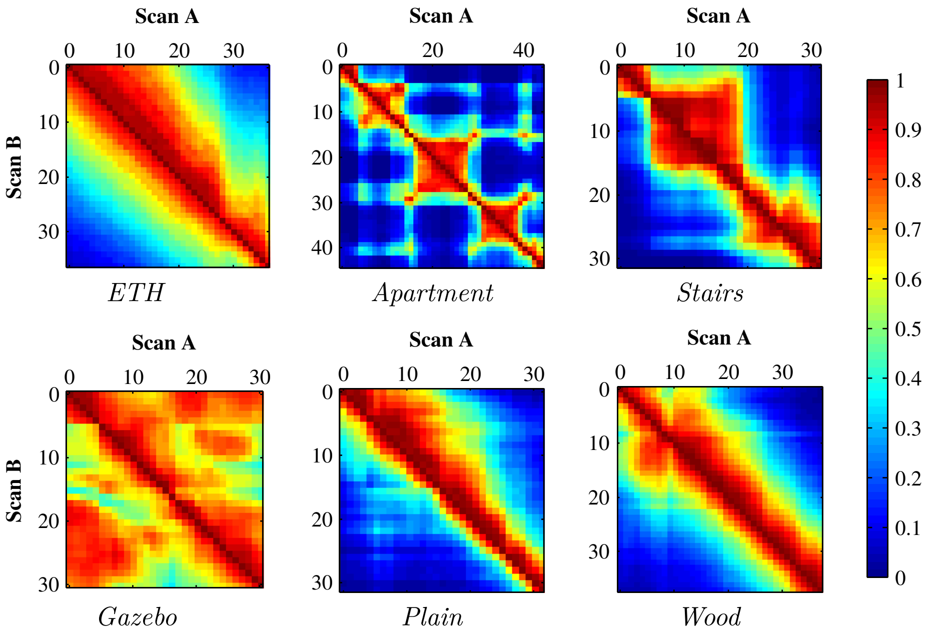


Figure 3.9: Estimated overlap for all data sets. Tables can be read as the ratio of point in Scan A that are also in Scan B. Dark red is high overlap and dark blue is low overlap. Diagonal elements have a ratio of 1. *Gazebo* in winter and *Wood* in summer were selected as representative of the duplicated data sets.

information specific for each data set. Photographs were also taken and marked on the topological map to give the user a quick overview of the real environment. Figure 3.10 shows an example of such representation. A graph of the platform orientations, gravity vectors, number of satellites and detected magnetic north vectors are also offer to the scientific community.

3.1.10 Data Sets for Triangulation Sensors

The later eight sequences cover a vast range of problematic linked to registration. For uniformity, only a Hokuyo UTM-30LX was selected as representative of time-of-flight sensors. However, sensors recovering depth based on triangulation have a different noise characterization and sensor noises can also be a predominant factor dragging down the quality of registration. We also wanted to tackle problems of fast motions and high data streaming rate. To evaluate those elements, we employed a Kinect sensor in a room equipped with a Vicon tracking system. The later provides ground truth position in the order of millimeter. We acquired several sequences under

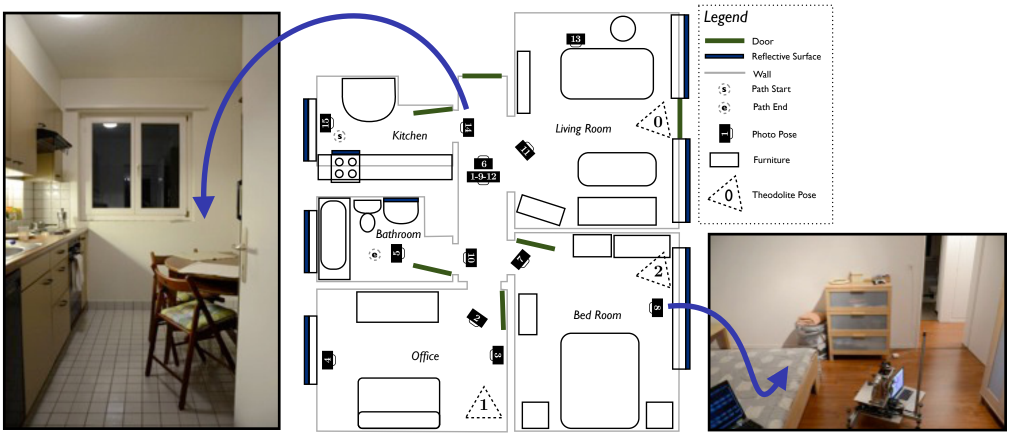


Figure 3.10: Example of contextual information: topological map of *Apartment* with associated photographs of the environment.

ROS [Cousins et al. 2010], using the Kinect OpenNI driver[1] and `rosbag` to record the data.

In their comparison of ICP performance, Rusinkiewicz and Levoy [2001] used three synthetic environments composed of low-frequencies, all-frequencies, and high-frequencies surfaces with some added noise. We reused this concept and transposed it in a real indoor experimental setup. We assembled three different static environments of increasing complexity (Figure 3.11). For each complexity, an operator performs three types of motions: (1) translations on the three axes (for about 10 s per axis), (2) rotations on the three axes (for about 10 s per axis), (3) a free fly motion over the scene (for about 15 s). We performed each type of motions, for all environments, at three different speeds: (1) slow motion with speed in the range of indoor ground robots (around 0.3 m/s), (2) medium motion with speed in the range of agile robots (around 0.5 m/s), (3) fast motion with a more arduous speed (around 1.3 m/s).

This gave us 27 sequences with point clouds produced by the Kinect at 30 Hz and its pose tracked by the Vicon at 100 Hz. We used a resolution of 160×120 depth pixels to generate the point clouds, which creates clouds containing at most 19200 points, as some points from the sensor were invalid.

[1] `http://www.ros.org/wiki/ni`

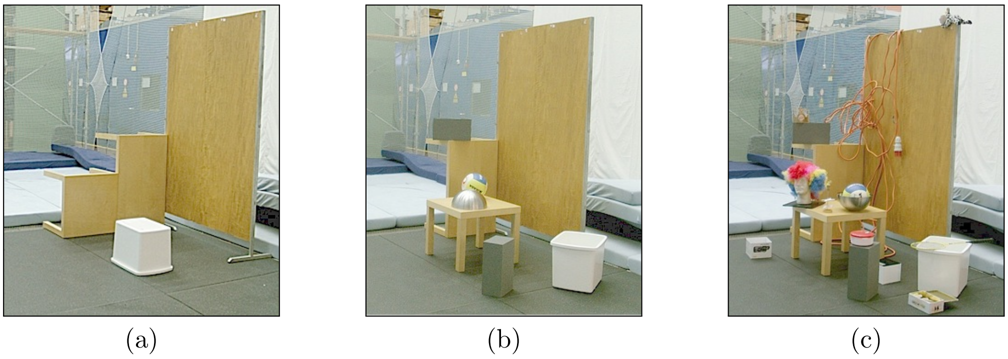


Figure 3.11: Experimental environments of (a) low complexity, (b) medium complexity, and (c) high complexity

3.2 Evaluations Protocols

There are multiple ways to compare ICP algorithms. We wanted to gather in a coherent protocol what researchers occasionally evaluate sparsely in their publications. This protocol proposes a list of important elements to compare against each other and to take into consideration in the interpretation. We observed that, even though a solution is developed to address a specific problem, the impact of the new solutions in a general context is often forgotten. Moreover, a common protocol allows faster comparison between ICP variants. This protocol encompasses an experimental methodology and evaluation metrics, as it is already proposed in other fields such as stereo correspondence detection [Scharstein and Szeliski 2002], multi-view stereo reconstruction [Seitz et al. 2006] and optical flow computation [Baker et al. 2007].

In this section, we highlighted the different elements that influence the outcome of ICP variants and what can be controlled in order to evaluate those variants. We also introduced robust metrics that we took into account for a quantitative assessment of the algorithm quality. The protocol was split into two types of evaluation: (1) *intrinsic*, referring to the stability of the solution against its own parameters, and (2) *extrinsic*, the robustness of the solutions against external inputs.

3.2.1 Intrinsic Evaluations

Solution characterizations are considered as intrinsic evaluation of the solution. Most of the solution have predominant parameters that need to be adjusted depending of the context of the application. Although those parameters are specific to a given solution and require specific setups to highlight their pertinence, some guidelines should be followed to help other researchers to build upon their work. Observations over a continuous range of parameters typically offer a deeper understanding of the solution behavior. This help to asses how critical is the parameter and can provide some insights on how to tune this parameter in another context. Even though the proposed solution can be very flexible in different situations, parameters need to be fixed for an evaluation against external inputs. To allow reproducibility, an explicit list of the parameters with their optimized value is required. Those optimized value should be the outcome of the solution characterization.

3.2.2 Extrinsic Evaluations

Although robotic platforms and applications have a long list of distinctions, we brought back those differences to three main input elements: (1) initial alignment, (2) overlap between scans and (3) types of environmental structure.

The ICP takes two scans for input with an initial alignment of one with respect to the other. As ICP is an approximate algorithm essentially doing local convergence, its result depends on the initial pose. This initial guess is typically provided by inertial-measurement accumulation, odometry or heuristic motion models, which all have limited precision and larger uncertainty with an increasing time between observations. It is therefore important to assess how well an ICP solution converges close to the correct pose based on various initial hypotheses. To this aim, we sampled the space of initial alignment by adding perturbations to a ground-truth value. While the error distribution of odometry models is usually not Gaussian for non-linear kinematic models, the deviation from a Gaussian depends on the actual model and command history, which goes beyond the scope of our data sets. As a reasonable approximation, we sub-sampled the perturbations from zero-mean 6D multivariate Gaussian distribution.

For the sampling of the initial poses, we designed three different sets of initial perturbations sampled from Gaussian distributions with three different variance magnitudes: *Easy*, *Medium* and *Hard*. We selected the variance for each of those sets with respect to the scale of the data sets. Figure 3.12 depicts two reference (human and scanner) to show the scale of the perturbations and presents a 2D projection of the the translation components. Figure 3.13 shows the cumulative probability as a function of translation error for the three perturbation sets(i.e., *Easy*, *Medium* and *Hard*). The filled backgrounds show the respective theoretical distributions.

It is worth noting that the norm of multivariate-Gaussian–distributed variables is an χ-distribution. The difference and the jaggedness of the sampled distribution compared to the theoretical distribution is due to the relatively low number of samples (64) compared to the six dimensions of the sampling space. As we aimed at proposing those perturbation samples to the community to allow everyone to compare their solution in the same conditions as ours, we felt that increasing significantly the number of perturbations would deter people from trying due to the computation time it would take. The sub-sampling we used required 2,240 tests per perturbation type per environment, which we consider to be a reasonable compromise between the number of samples and the evaluation time.

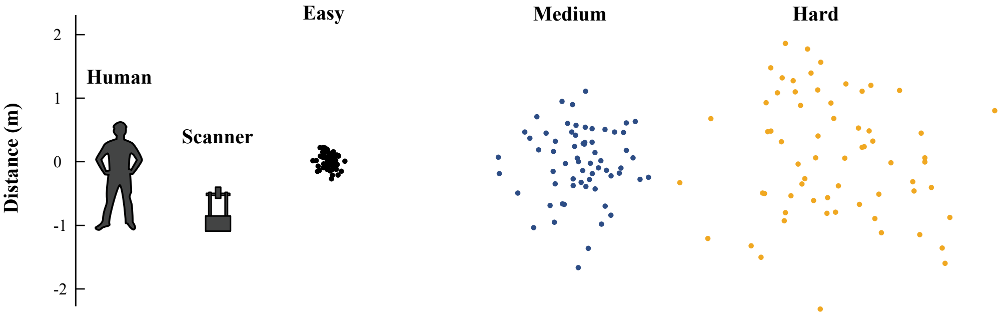


Figure 3.12: Overview of the scale for the 64 selected perturbations generated from Gaussian distributions: *Easy* (0.1 m, 10°), *Medium* (0.5 m, 20°) and *Hard* (1.0 m, 45°).

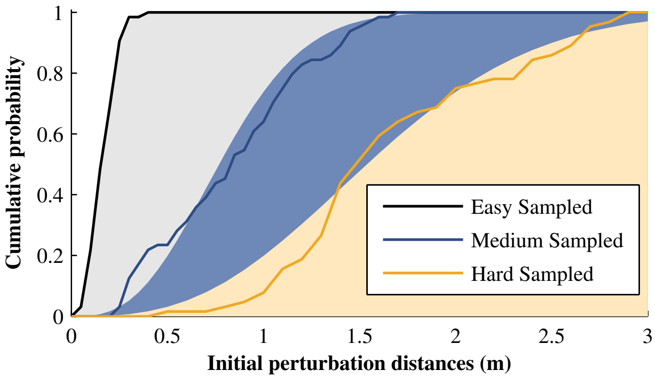


Figure 3.13: Cumulative probability of the initial perturbation distances for each of the perturbation sets. The lines are based on the actual 64 samples perturbations; the filled backgrounds correspond to the theoretical curves. The *Easy* sampled and theoretical curves overlay due to scaling.

The second factor driving the difficulty of scan matching is the amount of outliers. If there are a lot of points that do not correspond to the same features in both scans, the ICP runs the risk of converging to a local optimum driven by false matches. We quantified this phenomenon by assessing the overlap ratio of a scan with respect to another, as the outlier ratio is the complement of the overlap ratio. More formally, the overlap ratio is defined by the ratio of points of a *Scan A* for which there is a matching point in a second *Scan B*. In robotics, this overlap ratio is primarily governed by the field of view and the motion of the sensor. Indeed, without dynamic elements in the scene, the overlap ratio corresponds mainly to the ratio between the intersection of sensor fields of view on the one hand, and the field of view of the reference point cloud on the other hand. If the motion, especially for rotation, is large when compared to the field of view, then the overlap can be too low for the ICP to converge properly. For slow sensors, like 2D laser scanners generating 3D point clouds by rotating around an axis, it is therefore preferable to apply a scan matching for each consecutive pair of scans. However, on faster sensors like RGB-D cameras running up to 30 Hz, it is often possible, and even desirable, to skip several scans, as long as the overlap is not too low.

The quality of registration is very sensitive to overlap [Pathak et al. 2010b]. However, overlap is not homogeneous in a given data set path. For example, Figure 3.14 shows the evolution of the error in the *Apartment* data set for the point-to-plane distance metric. Scans were registered following the path, which means that every scan was paired with the scan recorded just before. In most cases, the registration is satisfying. However, there are a few places, around openings, where the registration performance degrades. Those places correspond to opening of the field of view which corresponds to a sudden decrease in the overlap. A change in overlap does not appear uniformly in all paths executed while recording the data sets. Thus, it is possible that the difference in overlap between two paths shade the impact of the type of environment. To overcome this limitation, we randomly selected 35 pairs of scans, ensuring a uniform coverage of the overlap between 0.30 and 0.99 for all data sets. Those pairs were selected using the values of Figure 3.9 with the lower bound of 0.30 forced by the lowest overlap value in *Gazebo*.

Finally, the environmental structures recorded in the scans themselves can have a huge influence on the registration quality. Indoor environments typically exhibit a lot of planar surfaces (e.g. ground, walls, ceiling, tables) that are therefore locally regular. In that case, if the matching step is slightly incorrect, a wrongly associated point still has a good chance of behaving like a correct point. On the other hand, natural environments with trees, bushes and herbs will have false matches that are detrimental to the error minimization. Moreover, environments without a reasonable

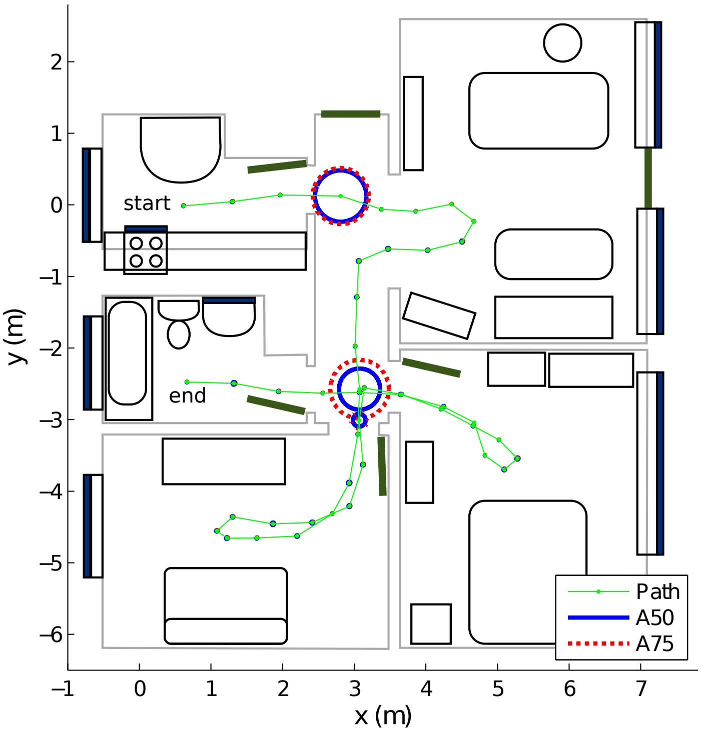


Figure 3.14: Point-to-plane solution in the *Apartment* data set: separate statistics for every pose. The path of the scanner (green) with the A50 and A75 quantile statistics overlaid on a sketch of the environment.

ratio of horizontal and vertical objects might lack information for a proper registration. This typically happens in long and straight hallways or outside on open spaces where the ground is the only major surface present. To capture the impact of the environment, we selected six different environments from the "Challenging Laser Registration" data sets described in Section 3.1. The selected data sets were: *Apartment*, *ETH*, *Stairs*, *Wood* (in summer), *Gazebo* (in winter), and Mountain *Plain*.

3.2.3 Evaluation Metrics

For each ICP solution, initial alignments (i.e. being the ground truth plus 64 perturbations) is applied to all 35 selected pairs of scans in a predefined environment. After all the registration are applied, the evaluation produces samples from the error distribution of resulting alignments for each pair of scans. Then, we proposed to use cumulated error distributions over all pairs of scans ease the interpretation of samples from that particular ICP solution for a given environment and a given perturbation level. We can also accumulate over the different environments for the marginal distribution of error of a given ICP solution.

However, this error distribution lies in $SE(3)$, the Special Euclidean group in dimension 3, whereas we are mainly interested in both the translation and rotation. Therefore, for simplicity, we projected the 6D distribution into the translation and rotation errors. Given the ground-truth transformation expressed by a 4×4 homogeneous matrix T_g and its corresponding transformation found by the registration solution T_r, we can define the remaining error ΔT as follows:

$$\Delta T = \begin{bmatrix} \Delta R & \Delta t \\ \vec{0} & 1 \end{bmatrix} = T_r T_g^{-1} \tag{3.1}$$

with its translation error e_t, defined as the Euclidean norm of translation vector Δt:

$$e_t = \|\Delta t\| = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{3.2}$$

and its rotation error e_r, defined as the Geodesic distance directly from the rotation matrix ΔR:

$$e_r = \arccos\left(\frac{trace(\Delta R) - 1}{2}\right) \tag{3.3}$$

Although those metrics are not invariant to rigid coordinate transformations (i.e., an error on rotation can also produce an error on translation), they uniform for every competitive solutions and represent quantity that can easily be interpreted.

In order to compare more easily these error distributions, we used robust statistics like the median and the quantiles instead of mean and covariance. Indeed, as the error distributions are far from Gaussians, the empirical mean and covariance are not really indicative values for interpreting precision and accuracy. This choice is similar to May et al. [May et al. 2009] where the authors defined A50, A75, A95 as the respective quantiles for probabilities 0.5 (i.e. the median), 0.75 and 0.95 of the error distributions. Another advantage of these statistics is that they allow interpretation in terms of accuracy and precision. The solution under evaluation is accurate if the values of A50, A75 and A95 are close to zero. The solution is precise if the difference between those quantiles are small.

We used a specific metric when evaluations need to be done on a sequence of scans, as opposed to pre-selected pairs of scan. As demonstrated by Kümmerle et al. [2009], it is better to evaluate the error between the relative poses instead of the error in a global frame of reference. In the case of the data set recorded with the Kinect, we had to select a significant relative displacement knowing that the data were recorded at 30 times per second. To provide robustness against noise, we cumulated the path over 30 registrations and then computed the error in translation e_t and in rotation e_r. This can be interpreted as a rate of error per second.

In sequential evaluation, it is possible that failures happen (i.e. low number of points, divergence, etc.), so it is preferable to output the same initial transformation to the system in order to not break the sequence of registration. In a situation where no motion model is used, this mean returning an identity matrix. In the case of registration applied at 30 Hz, this default behavior can be very close to the ground truth pose. Thus, it is important to keep track of those failures and penalize the evaluation accordingly. We kept track of the number failures N_{fail} over a dataset having a number of registrations N_{icp}. In the case of free-fly-motion sequences, $N_{\text{icp}} = 447$. In the case of translation and rotation sequences, $N_{\text{icp}} = 838$. We defined an ICP performance metric penalizing the number of failures as:

$$\text{perf} = \frac{N_{\text{icp}} - N_{\text{fail}}}{N_{\text{icp}}} \frac{1}{\text{median}(e_t)} \tag{3.4}$$

The first fraction gives the success ratio while the second one is the inverse of the median error of the data set. The intuition behind the use of this performance metric, instead of using directly the error, is that we expected that time and performance curves have similar trends. If the evolution of both curves follows the same tendency, it is difficult to devise a clear parameter optimum. With this ratio (Equation 3.4), the performance will be 0 if all registrations fail and will be equal to the inverse of the translation error if all registrations succeed. We can define a similar metric for rotation error by replacing e_t by e_r.

To have a deeper understanding of the error distribution, it is possible to use the cumulative function of this distribution instead of only the A50, A75 and A95 quantiles. Those graphs thus present the proportion of registration solutions that lie beneath a given error. Moreover, it is easy to see the value of this error for each quantile. This type of representation was called *Recall-Accuracy Threshold* in a previous work [Jian and Vemuri 2011]. An alternative representation of those results is the histograms of the number of registration solutions for each error bin, which corresponds to the derivative of the cumulative error that we proposed. However, that representation renders difficult the comparison of many error distributions and the depiction of the A50, A75 and A95 statistics.

Finally, the computing time can be an important factor, especially for online applications with real-time constraints and embedded systems with limited processing power. Thus the challenge is to obtain an absolute evaluation of the computing time that is relevant for different hardware and different use cases. The choice of the programming language, the technical level of the programmers, the amount of parallelism, etc., are all elements that could affect time performance. In general, time evaluation should be considered as qualitative measurement, unless all those elements are controlled and known to be as uniform as possible.

3.2.4 Protocol

With those metrics, we can now propose a protocol for the evaluation of ICP variants that goes beyond parameter identifications.

First, variants should always be compared to a commonly accepted ICP baseline. This contrasts with papers that compare novel variants between themselves in order to highlight a specific hypothesis. While we recognize the interest of these works, the amount of ICP variants presented in the literature calls for more effort to link them to each other.

Second, ICP variants need to be compared on enough data in order to reduce the risk of overfitting and to ensure statistically signifiant interpretations. Specific fields of applications may require specialized data sets, but efforts should be made to also compare over generic data sets. To obtain a comparison as unbiased as possible, the data sets should cover various kinds of environments at different overlap ratios. In this section, we have suggested to employ a group of 3D robotics data sets covering a variety of environments. Moreover, algorithms should be compared with different perturbation distributions in order to assess their robustness.

Finally, the comparison with other solutions should be made with respect to the distribution of errors rather than being made just on a single result.

In a nutshell, researchers evaluating their solution against the laser data sets should maintain a certain uniformity by:

1. Characterizing the main parameters of their novel solution.

2. Evaluating their solutions using predefined data sets, pairs of scans and perturbations.

3. Recording translation and rotation errors following Equation 3.2 and Equation 3.3.

4. Recording computational time excluding data acquisition but including preprocessing steps.

5. Reporting strength and weakness against environment type, perturbation level and overlap ratio.

6. Comparing their results with formal solutions in terms of precision and accuracy using A50, A75 and A95 statistics.

7. Making their results publicly available whenever possible, so that other researchers can accelerate the comparison process.

Although the later protocol should be used in priority, it might be more pertinent to realize the evaluation over more dynamic data sets. These data sets can highlight specific problems encountered by more agile and fast moving systems. In that case, we suggest to:

1. Characterize the main parameters of the novel solution.

2. Evaluate the solution using a predefined point cloud sequences.

3. Record translation and rotation errors following Equation 3.4.

4. Record computational time excluding data acquisition but including preprocessing steps.

5. Report strength and weakness against environment type, motion speed and motion type.

6. Compare their results with formal solution in terms of precision and accuracy using A50, A75 and A95 statistics.

7. Make their results publicly available whenever possible, so that other researchers can accelerate the comparison process.

3.3 Modular Registration Library

As explained in Chapter 2, the ICP is an iterative algorithm performing several sequential processing steps, both inside and outside its main loop. For each step, there exist several strategies, and each strategy demands for specific parameters.

To our knowledge, there is currently no software tool to compare these strategies. The Point Cloud Library (PCL) has a partial support for filters

Table 3.4: List of processing blocks available in `libpointmatcher`. This list displays the status of the library as of version 1.0.0 and is intended to evolve over time.

Modules	*Current Implementations*
Data filtering	`FixStepSampling`, `MaxDensity`, `MaxPointCount`, `MaxQuantileOnAxis`, `MinDist`, `ObservationDirection`, `OrientNormals`, `RandomSampling`, `RemoveNaN`, `SamplingSurfaceNormal`, `Shadow`, `SimpleSensorNoise`, `SurfaceNormal`
Data association	`KDTree`, `KDTreeVarDist`
Outlier filtering	`MaxDist`, `MedianDist`, `MinDist`, `SurfaceNormal`, `TrimmedDist`, `VarTrimmedDist`
Error minimization	`PointToPlane`, `PointToPoint`
Transformation checking	`Bound`, `Counter`, `Differential`
Inspection	`Performance`, `VTKFile`
Log	`File`

in its registration pipeline, but not a completely reconfigurable ICP chain[2]. To enable such a comparison, we have developed a modular ICP chain, as illustrated in Figure 3.15, and made it available as open source in the form of the `libpointmatcher` library[3]. This library is written in C++11, restricted to the subset supported by GCC 4.4 and more recent versions. In the ICP chain, every module is a class that can describe its own possible parameters, therefore enabling the whole chain to be configured at run time using YAML. This text-based configuration aids to explicit parameters used and eases the sharing of working setups with others, which ultimately allows for reproducibility and reusability of the solutions. Table 3.4 lists the available modules.

Our ICP chain takes as input two point clouds, in 2D or 3D, and estimates the translation and the rotation parameters that minimize the alignment error. We called the first point cloud the *reference* and the second the *reading*. The ICP algorithm tries to align the reading onto the reference. To do so, it first applies filtering to the point clouds, and then it iterates through a sequence of processing blocks. For each iteration, it associates points in reading to points in reference and finds a transformation of reading that minimizes the alignment error.

[2] We are in contact with PCL developers to integrate parts of our work into it.

[3] `http://github.com/ethz-asl/libpointmatcher`, version 1.0.0 at time of submission of this paper.

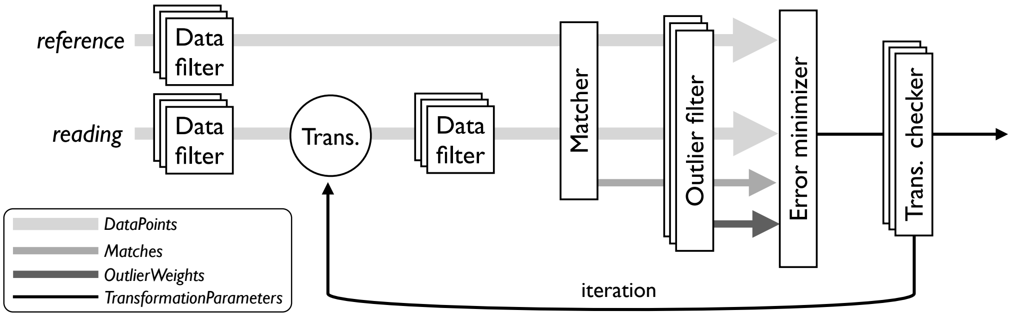


Figure 3.15: The modular ICP chain as implemented in `libpointmatcher`. Note that some data filters are applied to the reading only once, and some are applied at each iteration step.

3.3.1 Processing Blocks

More specifically, the ICP chain consists of several steps (Figure 3.15), implemented by modules (Table 3.4). The steps and the corresponding types of modules are:

- *Data filtering*: This step applies to both the reference and the reading point clouds. At this step, zero or more `DataPointsFilter` modules take a point cloud as input, transform it and produce another cloud as output. The transformation might add information, for instance surface normals, or might change the number of points, for instance by randomly removing some of them.

- *Transformation*: The reading point cloud is rotated and translated. Additional data, such as surface normals, are transformed as well.

- *Data association*: A `Matcher` module links points in the reading to points in the reference. Currently, we provide a fast k–nearest-neighbor matcher based on a kd-tree, using `libnabo`.

- *Outlier filtering*: Zero or more `OutlierFilter` modules remove (hard rejection) and/or weight (soft rejection) links between points in the reading and their matched points in the reference. Criteria can be a fixed maximum authorized distance, a factor of the median distance, etc. Points with zero weights are ignored in the subsequent minimization step.

- *Error minimization*: An `ErrorMinimizer` module computes a transformation matrix to minimize the error between the reading and the reference. Different error functions are available, such as point-to-point and point-to-plane.

- *Transformation checking*: Zero or more `TransformationChecker` modules can stop the iteration depending on some conditions. For example, a condition can be the number of times the loop was executed, or can be related to the matching error. Because the modules can be chained, we defined that the relation between modules must agree through an OR-condition, while all AND-conditions are defined within a single module.

3.3.2 Data Types

The ICP chain provides standardized interfaces between each step. This allows for the addition of novel algorithms to some steps to evaluate their effect on the global ICP behavior. These interfaces are:

- The `DataPoints` class represents a point cloud. For every point, it has features and, optionally, descriptors. Features are typically the coordinates of the point in space. Descriptors contain information attached to the point, such as its color, its normal vector, etc. For both features and descriptors, every point can have multiple channels. Every channel has a dimension and a name. For instance, a typical 3D cloud might have the channels "x", "y", "z", "w" of dimension 1 as features (using homogeneous coordinates), and the channel "normal" of size 3 as descriptor. There are no sub-channels, such as "normal.x", for the sake of simplicity. Moreover, the position of the points is in homogeneous coordinates because they need both translation and rotation, while the normals need only rotation. All channels contain scalar values of the scalar type from the template parameter. Although this might be sub-optimal in memory, it eases greatly the interaction between the different modules.

- The `Matches` class is the result of the data-association step, before outlier rejection. It corresponds to a list of associated reference identifiers, along with the corresponding *squared* distance, for all points in the reading. A single point in the reading can have one or multiple matches.

- The `OutlierWeights` class contains the weights of the associations between the points in `Matches` and the points in the reference. A weight of 0 means no association, while a weight of 1 means a complete trust in association.

- The `TransformationParameters` is a transformation in the Special Euclidean group of dimension n, $SE(n)$, implemented as a matrix of size $n + 1 \times n + 1$.

3.3.3 Implementation

All modules are children of parent classes defined within the `PointMatcher` class. This class is templatized on the scalar type for the point coordinates, typically `float` or `double`. Additionally, the `PointMatcherSupport` namespace hosts classes that do not depend on the template parameter. Every kind of module has its own pair of *.h* and *.cpp* files. Because modules can enumerate their parameters at run time, only the parent classes lie in the publicly accessible headers. This maintains a lean and easy-to-learn Application Programming Interface (API).

To use `libpointmatcher` from a third-party program, the two classes `ICP` and `ICPSequence` can be instantiated. The first provides a basic registration between a reading and a reference, given an initial transformation. The second provides a tracker-style interface: an instance of this class receives several point clouds in sequence, and continuously updates the transformation with respect to a user-provided point clouds. This is useful to limit drift due to noise in the case of high-frequency sensors [Pomerleau et al. 2011b]. A common base class, `ICPChainBase`, holds the instances of the modules and provides the loading mechanism.

While developing novel solutions, it is crucial to understand what is going on, in particular in complex processing pipelines like the ICP chain. Therefore, `libpointmatcher` provides two inspection mechanisms: the logger and the inspector. The logger is responsible for writing the information during the execution to a file or to the console. It will typically display light statistics and warnings. The inspector provides a deeper scrutiny than the logger. There are several instances of inspectors in `libpointmatcher`. For instance, one dumps ICP operations as VTK files [Schroeder et al. 2006], allowing to visualize the inner loop of the algorithm frame by frame. Another inspector collects statistics for the performance evaluation.

3.4 Summary

In this chapter, we introduced new data sets covering a diversity of challenging environments for registration algorithms. Although some of those environments can be found in already available data sets, our *Laser Registration Data Sets* englobe them all in a coherent group recorded with the same methodology and material. We achieved precise localization of the scanner using a theodolite, which gave us the ability to record data sets in GPS denied environments, indoors or outdoors, with the same setup. The precision achieved is also higher than when using data sets that are already available to the community, which should ease the evaluation of registration algorithms on a fair base. This will allow for the development of improved registration algorithms when mapping challenging environments, such as found in real world situations. The data sets and supporting information are publicly available under the section *IJRR - Laser Registration Datasets* at:

http://projects.asl.ethz.ch/datasets

Complementary data sets for triangulation sensors with fast motions is also available under the section *IROS 2011 - Kinect Dataset*.

The protocol proposed focus on comparing ICP variants but the guidelines could be applied to most of the registration evaluations. The results obtained using the *Laser Registration Data Sets* will be made available through the same web site. This should allow researchers to produce comparative results without having to re-implement other solutions.

We also presented an open-source modular ICP library that can further improve the repeatability by allowing easy tests and comparisons with baseline variants. Thus, this modular library is the companion of choice of our protocol. The library comes with code examples, documentation and wrapper for Robot Operating System (ROS). It can be found on the public repository:

https://github.com/ethz-asl/libpointmatcher

Online tools have also a strong influence in older research fields. For example in biology, GenBank is a database keeping track of which organisms have their DNA sequenced. This avoid multiple publications of the same genome.

Publications are an essential tool for scientific research, but online tools can also help gathering knowledge on a specific topic. When it comes to hardware, repeatability of an experiment can be tedious to achieve. On the contrary, softwares does not have the same physical limitations. ROS and PCL are good examples of open source softwares having an impact in the robotics field. The drawback is that it takes extra work to document well to make the information accessible publicly and to maintain the information up-to-date. In this work, we invested time to ensure that the data sets, the evaluations results and the softwares are publicly available, hoping to support further research. We believe that the combination of data sets, protocol and software nicely shows how open-source softwares can drive research forward.

Chapter 4
Characterizations and Evaluations

Registration solutions applied to robotics have their own needs and constraints. Even if ICP algorithms are 20 years old, many systems fail due to a lack of knowledge on intrinsic and extrinsic characteristics of novel algorithms. In this chapter, we provide a deeper analysis of typical sensor noises, real-time solutions and robustness against a range of different inputs for improving successful autonomous system heavily relying on registration algorithms.

Common time-of-flight sensors used in robotics are still missing noise quantification. Section 4.1 presents noise models for three types of laser and discusses of the impact of the noise on surface reconstructions. Those results were presented at the *2nd International Conference on Applied Robotics for the Power Industry* [Pomerleau et al. 2012a]. The problem of real-time optimization is addressed in Section 4.2 and conclusions were published at the *IEEE 2011 International Conference on Intelligent Robots and Systems* [Pomerleau et al. 2011b]. Modern mobile platforms are expected to localize themselves in a range of environments. In Section 4.3, we approach the question of how viable are the original solutions proposed by Besl and McKay [1992] and Chen and Medioni [1991]. This refreshes the observations realized by Rusinkiewicz and Levoy [2001], which were mainly based on simulated data. The main conclusions of our evaluations were accepted for publication in the journal of *Autonomous Robots* [Pomerleau et al. 2013]. The chapter concludes in Section 4.4 with some guidelines for tuning ICP solutions according to the application.

4.1 Sensor Noise Identification

Many studies propose rangefinder sensor characterizations evaluating depth measurement errors under one or multiple conditions. A first group of characterizations evaluates the depth readings with respect to the target distance, incidence angle, brightness/reflectivity and heat level (radiance). Other type of studies evaluate the robustness of measurement in different ambient lights and air conditions (e.g., dust, rain, smoke, etc.). Finally, range sensors were found to be sensitive to their internal temperature, which can evolve in time, especially during the first hour of utilization. This type of error can easily be in the range of centimeters [Kneip et al. 2009]. The mixed pixel problem (i.e., a single laser beam footprint large enough to cover two objects at different depth) is first highlighted by Hebert and Krotkov [1991] and is still present in more recent sensors.

The most studied scanner is the Sick LMS-200, which is first characterized by Ye and Borenstein [2002], and further investigated in multiple work [Alwan et al. 2005; Jain et al. 2011; Pascoal et al. 2008; Lee and Ehsani 2008; Sanz-Cortiella et al. 2011]. A new generation of smaller Sick laser (LMS-100 family) is evaluated by Rudan et al. [2010]. This new scanner was presented as a competition to the smaller scanners proposed by Hokuyo. Evaluations for the Hokuyo series are well detailed with the characterization of the PBS-03JN [Alwan et al. 2005], the UBG-04LX-F01 [Park et al. 2010], the URG-04LX [Pascoal et al. 2008; Kneip et al. 2009; Paul et al. 2009; Okubo et al. 2009] and, more recently, the UTM-30LX [Tretyakov and Linder 2011; Li et al. 2011; Hrabar 2012]. Comparisons between Sick LMS-200 and Hokuyo URG scanners is also tackled in the literature [Lee and Ehsani 2008; Pascoal et al. 2008]. At a larger scale, Wong et al. [2011] compare 10 sensors, from the Faro Photon80 up to a custom made stereo ring, in an underground mapping situation. Time-of-flight cameras, from Mesa Imaging (formally SwissRanger), are characterized by Kahlmann et al. [2006] and May et al. [2009], leading to calibration models. Calibration for the Velodyne is also proposed by Atanacio-Jimenez et al. [2011] to improve the accuracy of the scanner from 23 to 1.5 cm. Recently, the Kinect is used in many robotics applications and is also characterized by Khoshelham and Elberink [2012]. Those studies, for sensors that are not discontinued, are regrouped in Table 4.1 to ease further consultation.

Based on Figure 2.1 and Table 4.1, we selected sensors with a small enough weight to be carried by medium-size robots, while ensuring a reasonable scanning range. We focused on time-of-flight and phase-shift depth sensors to ensure uniformity in the evaluation methodology. We investi-

	UTM	*Kinect*	*UBG*	*PBS*	*LMS 200*	*URG*
Target						
distance	1	2	3	4	5, 6	5, 7, 8
brightness	–	9	3, 9	4	5, 6	5, 7, 8
reflectivity	–	–	3	4	5, 6	5, 7, 8
incidence angle	–	–	–	4	5, 6	5, 7, 8
heat	–	–	–	–	5	5
Ambient						
conditions	1, 10	10	–	–	5	5
lights	–	–	–	–	–	8
Other						
time/temp. drift	–	–	3	4	6, 5, 11, 12	7, 8, 5, 12
mixed pixel	1	–	–	4	6, 13	7
sensor attitude	–	–	–	–	–	8

Table 4.1: Related studies of popular range sensors in relation with different types of evaluations. 1: [Li et al. 2011], 2: [Khoshelham and Elberink 2012], 3: [Park et al. 2010], 4: [Alwan et al. 2005], 5: [Pascoal et al. 2008], 6: [Ye and Borenstein 2002], 7: [Okubo et al. 2009], 8: [Kneip et al. 2009], 9: [Park et al. 2011], 10: [Tretyakov and Linder 2011], 11: [Jain et al. 2011], 12: [Lee and Ehsani 2008], 13: [Sanz-Cortiella et al. 2011].

gated the UTM-30LX and the LMS-151 to provide uncertainty models for which no models were found in the literature. Finally, we also studied the URG-04LX with the same methodology to compare our results with prior characterizations [Kneip et al. 2009; Okubo et al. 2009; Pascoal et al. 2008].

4.1.1 Experimental Protocol

A first set of measurements has been produced to determine the width of the laser beams. This beam deviation is present in the specifications of the SICK lasers but not in the Hokuyo ones. Each sensor scanning plane was put perpendicular to a wall at four different distances: 1, 3, 5 and 9 m. For each distance, a picture was taken using a camera without an infrared filter. A ruler fixed on the wall was used to determine the beam diameter as shown in Figure 4.1.

Moreover, a larger set measurements were recorded to evaluate depth error. As opposed to most protocols used in formal characterization, which focus on a single beam, we realized our evaluation on full 3D scans of metallic plates. We believed that such type of evaluation is much closer to real applications and leads to more realistic models. We used the custom-

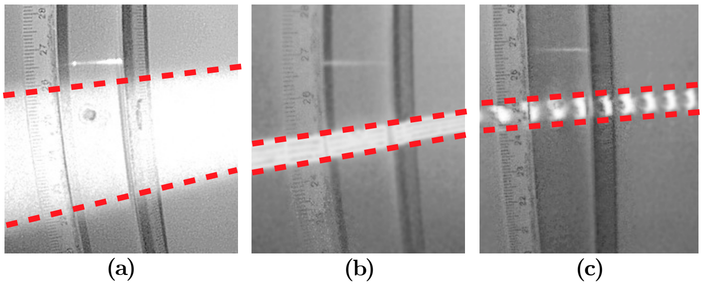


Figure 4.1: Different laser patterns detected using an infrared camera with the laser at 3 m: (a) LMS-151, (b) UTM-30LX and (c) URG-04LX.

made tilting platform, described in more details in Section 3.1.3, to mount the 3 different lasers and take 3D scans of different surfaces. The surfaces have been selected to represent a subset of metallic surfaces. We aimed at characterizing harsh and normal conditions to ensure that we did not under evaluated the noise. The surfaces were roughly 2 m long per 1 m large and the materials were: aluminum, metallic surface covered with paint (white board usually used with colored markers), old steel and rusty iron. Figure 4.2 presents photographs of all plates.

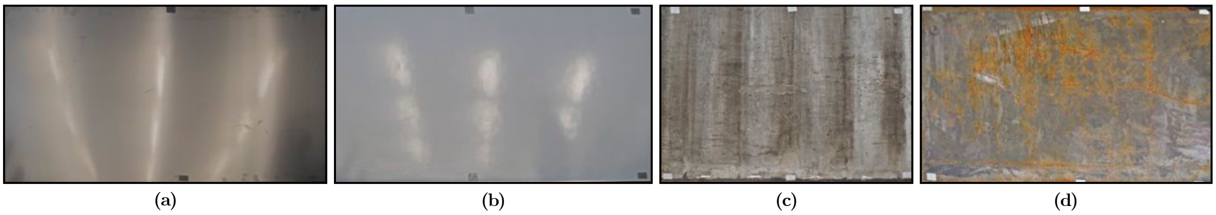


Figure 4.2: Scanned plates: (a) aluminum, (b) white board, (c) steel and (d) rusted iron.

We scanned those surfaces at five different distances, namely 0.2, 1.0, 2.0, 4.0 and 8.0 m. The theodolite TS15, from Leica Geosystem, was used to collect ground truth distances in a precision range of millimeters. The poses (i.e., orientations and positions) of the scanner were determined with the same protocol as in Section 3.1.3. As for the plate poses, six markers were installed on the periphery of the plates and measured with the same theodolite. Given that we obtained the transformation from the theodolite to the scanner, we can also express the points on the plates in the reference frame of the scanner. Using the location of those markers, the plate is divided in four triangles against which all points from the 3D scan of the plate are segmented (Figure 4.3 - left). With some basic geometry concepts, the error of each beam was determined given the triangle in which it is associated (Figure 4.3 - center). For each 3D scan, the intensity reflected by the surface was also recorded (Figure 4.3 - right). Those measurements were realized indoors for all combinations of sensors, materials and distances. The experimental setup was repeated outdoors for the the UTM-30LX to observe the influence of the sun on the measurements. The illumination of the environment at the plate level was measured with the TES 1332a Digital LUX Meter. This gave us a total of 80 different 3D scans with an average of 30,000 points per scan.

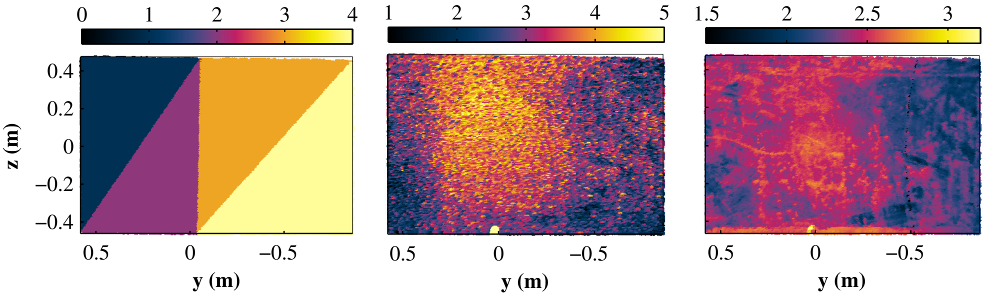


Figure 4.3: Example of the results based on a scan of the iron plate at 1.0 m using the UTM-30LX. Left: point segmentation based on ground truth measurements, color representing the clusters. Center: depth error [mm] on each beam value (positive when too close and negative when too far from the plane), color representing error. Right: intensity [10^3] returned for each beam, color representing the intensity as outputted from the UTM.

4.1.2 Random noise

For all results reported in this subsection on noise modeling, we removed measurements that were heavily affected by reflection. In our data set, those points were producing errors larger than 0.05 m. We separated the evaluation of those spurious readings and addressed them in Section 4.1.2.

We first analyzed the angular uncertainty for each laser. We used a linear regression in the form of $y = ax + b$ to evaluate the beam opening angle (Figure 4.4). For all fittings, the coefficient of determination R^2 was larger than 0.95. Based on those results, we confirmed the manufacturer specifications for the LMS-151 with an opening angle of 0.83 °. For the URG-04LX and the UTM-30LX, we evaluated the angles to be 0.13 ° and 0.14 ° respectively.

For the uncertainty on depth, we used the measurements realized over different distances and clustered them per sensor and material as depicted in Figure 4.5. For every type of material, represented with letters in the Figure,

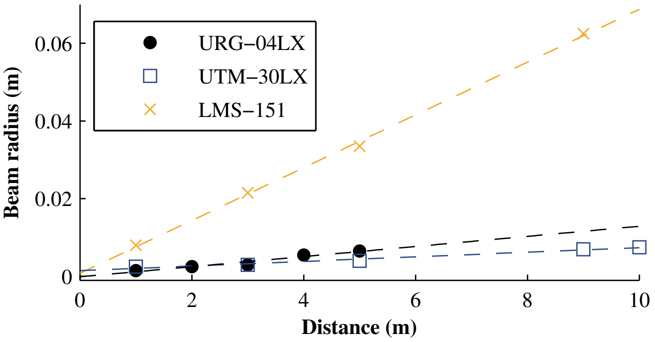


Figure 4.4: Radius of the laser beam with respect to measurement distances. Dots represent the experimental measurements and the dashed lines the modeled beam radius.

the median is marked with a point while the 50% of the measurements are contained within the error bars. Although systematic noise can be observed, especially for the URG-04LX given the different materials, we were mainly interested in the disparity of the depth measurement error for the model. Considering that all types of materials are equally likely to be found in the environment, we used the mean of the distances between each error bar as an estimate of the STD. This leads to a disparity of 0.028 m for the URG-04LX, 0.018 m for the UTM-30LX, and 0.012 m for the LMS-151.

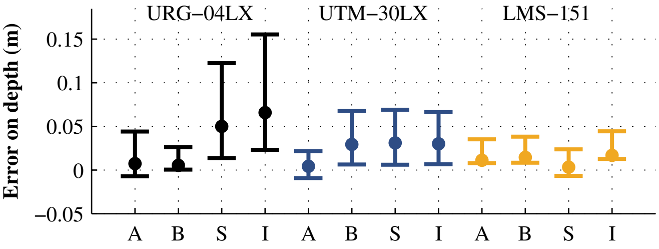


Figure 4.5: Error in depth measurement for different types of material (A: Aluminum, B: White board, S: Steel, and I: Iron).

We defined 2 types of error models for every point based on the depth returned by the sensors. Both models approximate the dispersion using Gaussian representation with the mean being the point read. The first model is anisotropic and is parametrized with a vector representing the beam direction $\vec{b}$ supporting the STD on depth σ_d. The STD of the beam radius σ_r is supported implicitly by any vector perpendicular to $\vec{b}$. The second model is a further simplification using an isotropic representation with only one STD defined as $\sigma_m = max(\sigma_d, \sigma_r)$. Figure 4.6 presents the notation and the comparison between the two models.

Parameters estimated based on our characterization are summarized in Table 4.2, for all sensors. A graphical comparison of the isotropic noise model for the three rangefinder lasers is depicted in Figure 4.7. We added the Kinect noise model from Khoshelham and Elberink [2012] for comparison. One

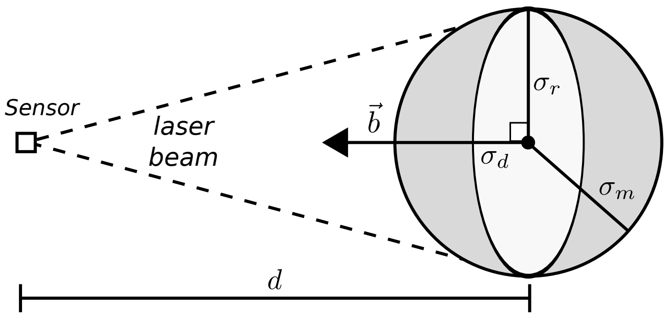


Figure 4.6: 2D projection of the isotropic (gray) and the anisotropic (light gray) error model. In 3D, σ_m corresponds to the radius of a ball while σ_r is uniform around $\vec{b}$

should be careful when comparing the LMS-151 and the Kinect because the largest uncertainty of the Kinect is mainly observed on depth measurements while the LMS-151 uncertainty is mainly caused by the laser beam opening angle when the depth is larger than 1.6 m. One can also notice that more precise measurements can be obtained with the Kinect if the expected structure is within 4 m. Otherwise, the UTM-30LX should be used. On one side, the opening angle of the LMS-151 produces less precise measurements at long range. On the other side, this opening angle coupled with its angular resolution ensures an overlap of 50% of every beam by the subsequent one, giving more safety about the reading. The LMS-151 also has a longer range than the UTM-30LX.

Table 4.2: Uncertainty model parameters for the three lasers studied. The measured depth d is expressed in meters.

Sensor	*Anisotropic*	*Isotropic*	
URG-04LX	$\sigma_r = \frac{1.3d-0.1}{1000}$	$\sigma_m = 0.028$	
	$\sigma_d = 0.028$		
UTM-30LX	$\sigma_r = \frac{0.6d+1.48}{1000}$	$\sigma_m = 0.018$	
	$\sigma_d = 0.018$		
LMS-151	$\sigma_r = \frac{6.8d+0.81}{1000}$	$\sigma_m = 0.012$	if $d < 1.646$
	$\sigma_d = 0.012$	$\sigma_m = \frac{6.8d+0.81}{1000}$	else

When comparing our model for the URG-04LX to other studies, two main differences appear. First, we did not adjust the depth measurements based on a linear [Okubo et al. 2009] or third order polynomial [Kneip et al. 2009] correction. Based on our measurements, we could not find a simple model that was only based on the measured depth that could be useful in

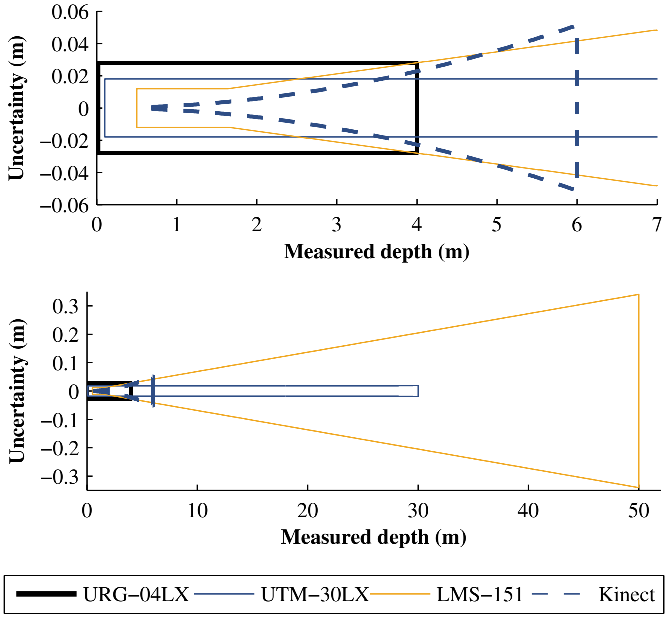


Figure 4.7: Comparison of the isotropic error model for the three lasers studied compared with the error model of the Kinect from [Khoshelham and Elberink 2012]. Top: close-up of the models at short range. Bottom: models up to the maximal range of all sensors.

most environments. Second, our STD is roughly ten times larger than the ones suggested in the other studies. We argue that formal STD reported are underestimated mainly due to the methodology used. The other evaluations take into consideration a single beam on an optimal material (medium gray [Okubo et al. 2009] or white paper [Kneip et al. 2009]), without considering the impact of the incidence angle. When looking more carefully at the larger range of error they reported over different incidence angles and materials, one can easily observe a variance in the order of centimeters, which seems consistent with our model.

It might be surprising that we went for a constant noise model but, as demonstrated in Figure 4.8, the simple model encompass well the random noise and avoid the risk of over fitting a function with high degrees of freedom. It is important to note that for each cluster of distances, we subtract the median to reveal the variance of the data. The median error, on the other side, is dependent on distance, reflectivity of the surface, angle of incidence, etc.

Sensitivity to structure extraction

As explained in Section 2.3, structural information plays a critical role in the registration process. Although differential geometry proposes useful theories about shape modeling, noise in the reading might limit their usabilities. A

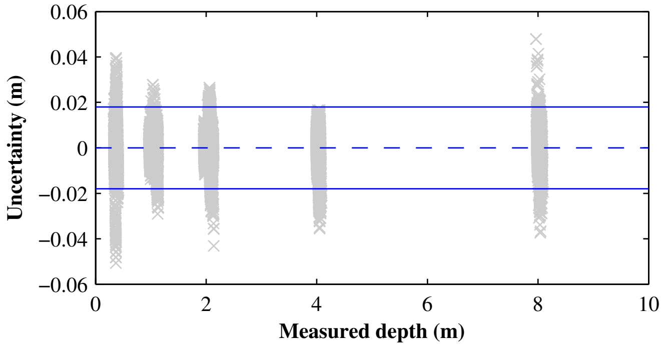


Figure 4.8: Raw variance results for the UTM-30LX. Gray points represent variance measured, while the blue lines the model proposed.

typical way to extract surface normal vectors from a point is to search for its NN, then recenter this subset to their mean and finally select the Eigen vector associated to the smallest Eigen value as the surface normal. On one side, the maximal radius for which the NN search is bounded depends on the size of the expected planar structure. The larger is the radius, the more robust to noise the extracted planar surface will be. On the other side, the lower bound for the NN search radius depends of the expected noise of the sensor used. We characterized this sensitivity to noise by extracting surface normal for each point, i.e., all sensors, all distances and all materials, and variated the size of the NN search radius, $r = \{0.01, 0.05, 0.1, 0.2, 0.5\}$ m.

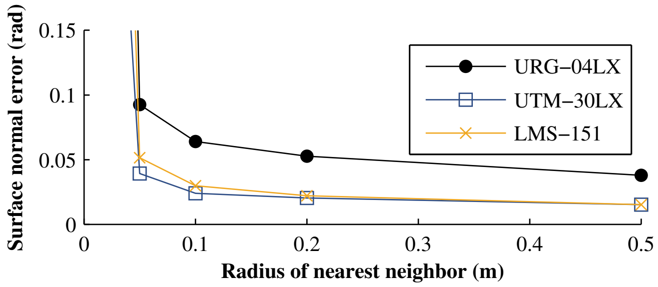


Figure 4.9: Medians surface normal errors (rad) for different NN search radius and the three sensors studied. Note that the errors reported for 0.02 m (> 0.6 rad) are cut to improve readability.

Figure 4.9 presents the medians of the resulting surface normal error (rad) for each sensor over the tested NN radius. Given that the evaluation was done on large planar surfaces, one can notice that the median error slowly decreases with a larger radius. Also, the median error grows very quickly under 0.05 m for all sensors. Based on those observations, we propose this value as the minimum radius that should be used for surface extraction. Within that lower bound, we can expect to have a surface normal error around 3.5° (0.06 rad) for the URG-04LX and around 1.6° (0.03 rad) for the UTM-30LX and the LMS-151. Of course, those values will hold only in the case where the density of point is high enough so the noise can be minimized.

Impact of sunlight

We observed a difference between the measurements taken indoors and outdoors, but the link between the sunlight and the difference in the reading is not that obvious. Prior evaluations concluded that the internal temperature of the sensor can influence greatly the reading (see Table 4.1 column: time/temp. drift for references). The main influence might arise from the sun heating the sensors. More controlled experiments must be conducted to confirm this hypothesis.

Impact of reflection and large intensity range

Reflecting surfaces like the aluminum plate poses three challenges. First, when the incidence angle is large, most of the energy is not reflected back to the sensor, which can lead to miss some measurements. Based on Figure 4.10, the URG-04LX seems to be more sensitive to this phenomena than the others. Second, there is also the probability that the beam gets reflected to another surface, leading to an overestimation of the depth. On Figure 4.10, the lower part of the plate displayed a larger error due to the ground being reflected. We concluded that the LMS-151 is more prone to such reflection given that the energy emitted by the sensor must be stronger to reach its maximal range (50 m). Finally, reflective plates exhibit a larger spectrum of reflected intensity, which seems to create systematic error that produces wave patterns. Preliminary results showed that there is a strong correlation between the error and the intensity, but deeper a investigation is required to propose a correction model.

This analysis gives us better understanding on noises that will be passed to the registration solution. Therefor, we can use the noise models in the error minimization function and relax the constraints induce by noisy points.

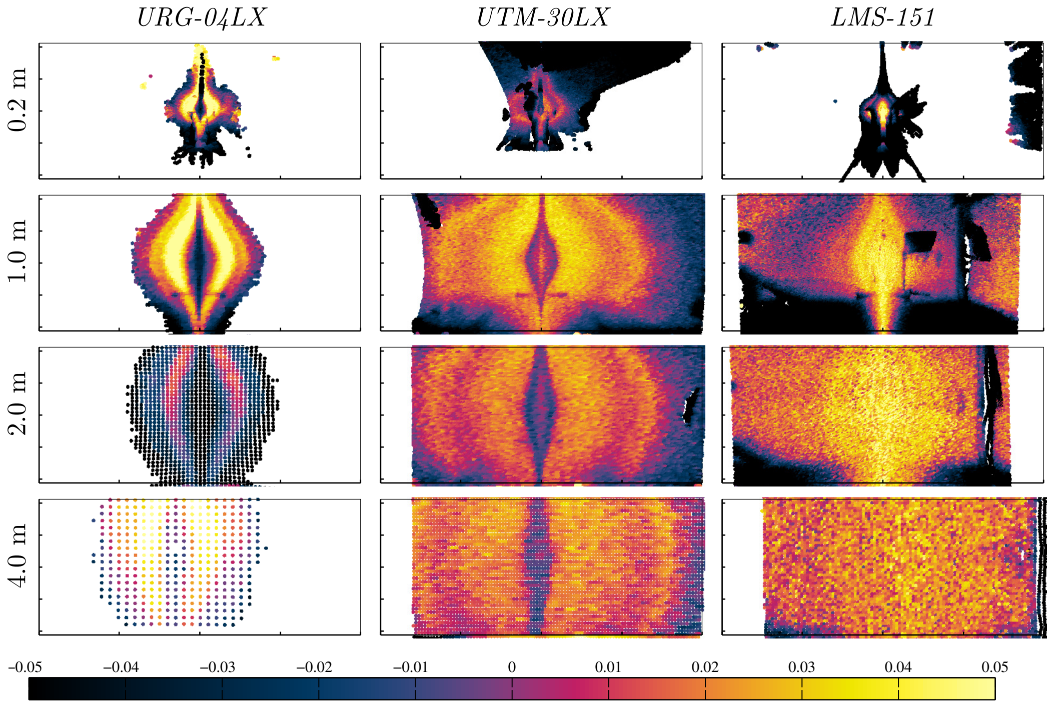


Figure 4.10: Front view of the aluminum plate for different distances. Color represents the error on depth with black being the error mainly due to reflection.

4.2 Fast Tracker Characterization

2D rangefinders were a cornerstone to the development of mapping and navigation field in the last two decades. Nowadays, rotating laser scanners, stereo cameras or depth cameras (RGB-D) can provide dense 3D point clouds at a high frequency. Large point clouds are costly to process so approximation methods were developed to accelerate registration speed. The chief assumption of ICP is that the association between points is mostly correct when using the closest point. If not, the computed transformation may be irrelevant. There are typically two ways to ensure that the association is correct: (1) attaching descriptors to the points to ease disambiguation, or (2) applying the ICP algorithm fast enough to limit the magnitude of changes. Descriptors are widely used in the vision community to match images and recently, 3D descriptors have been introduced to help the association step of ICP (see the work of Pathak et al. [2010b] or Zhuo and Du [2010] as recent examples). While the descriptor approach is promising, most elaborated geometric descriptors are still too computationally costly for online processing. In the section, we explored the second option.

Our goal was to optimize the use of lean and simple descriptors to produce an ICP-based 3D pose estimator at the frame rate of modern RGB-D sensors and to observe the impact on registration quality. Thus, we focused on simple solutions regarding sensor-noise modeling, point selection, and matching. We concentrated on the improvement of the tracker speed while keeping the pose estimation in a usable range. This was done by exploiting the modularity of our ICP library to adapt different filters. Finally, we showed statistical analysis of the tracker behavior in the context of indoor navigation using a Microsoft Kinect. We performed this evaluation using the data sets presented in Section 3.1.10.

4.2.1 Tracker

Using `libpointmatcher`, presented in Section 3.3, we implemented a fast tracker for the registration evaluation. This tracker takes as input a stream of point clouds and produces as output an estimation of the 6D pose of the sensor. To avoid drift, the tracker holds a single reference and matches every incoming point cloud against it. If the ratio of matching points drops below a pre-defined threshold, the tracker creates a new reference with the current cloud. This keyframe-based mechanism allows a higher frame rate by reducing the number of kd-tree creation and by limiting drift if the sensor stays at the same position. To easily explore the different parameters that affect the performance of the ICP algorithm, the ICP chain is completely configurable at run time.

4.2.2 Experimental protocol

A typical experiment on a data set implies a single value of time and performance over N_{icp} for a given parameter. Then, we repeated the computation N_{test} times to increase statistical significance, as some filters introduce randomness. We repeated these over a range of parameters N_{par} for different datasets. Such experimentation gives us a graph such as Figure 4.11 (left). Then, to ease the interpretation of results, we used robust estimators (i.e. median or quantiles as opposed to mean or variance) to extract the mode and the dispersion of the distribution for a given parameter. Figure 4.11 (right) shows the extraction of the median (A50) with the A75 and A95 quantiles. In our experiments, we observed that quantiles followed the same tendency as the median, so in further graphs we only presented the median for simplicity. Moreover, we made the assumption that a small parameter

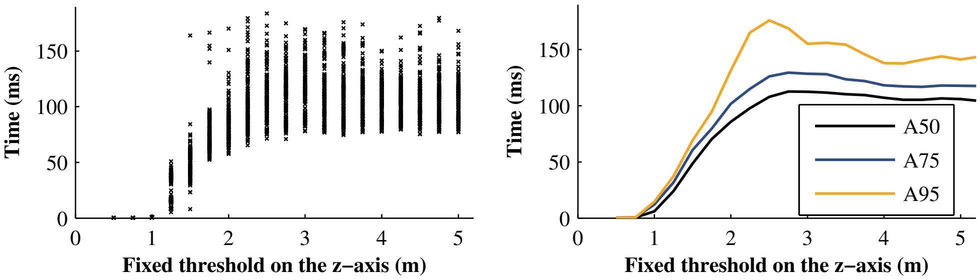


Figure 4.11: Example of processing taken from the sensor noise experiment (see the following *Sensor Noise* Section). Left: raw results. Right: extracted metrics based on 0.5, 0.75 and 0.95 quantiles.

change will be continuous over a larger range of situations that we can test. Following this assumption, we smoothed each curve using a moving average with a window size of 3.

We first explored parameters related to sensor noise, subsampling, and NN approximation. We used the data sets with the free-fly motion at low speed within the three types of environments. Given the resulting optimized parameters, we evaluated the robustness against all 27 data sets and also looked at the effect of the hardware on the processing speed. All these experiments used a different number of tests and parameters. Table 4.3 summarizes the configuration of each experiment, with the final column representing the total number of ICP computed per experiment, expressed as a factor of 1'000'000. The total number of registrations required for the experimental section is around 11 millions.

Experiment Names	N_{icp}	N_{par}	N_{test}	*Total (M)*
Sensor noise (fixed)	3×447	20	45	1.2
Sensor noise (ratio)	3×447	19	45	1.2
Subsampling (ratio)	3×447	39	30	1.6
Subsampling (step)	3×447	39	30	1.6
NN approximation	3×447	20	60	1.6
Robustness	9×447	1	20	3.8
	$+18 \times 838$			
Hardware speed	1×447	39	20	0.4
				11

Table 4.3: Number of ICP per experiment

Additionally, we fixed the error minimization solution as being the point-to-plane error [Chen and Medioni 1991], and the outlier filter being the median distance [Chetverikov et al. 2002] for all experiments.

Sensor Noise

The first experiment deals with methods to handle the sensor noise. Based on parallax, the Kinect has an accuracy on the depth that is inversely proportional to the distance. Moreover, it has a dead zone of 0.4 m close to the sensor. To deal with the Kinect noise, we explored two techniques: (1) a fixed threshold to prune points over a certain depth, and (2) a ratio of points to keep with the smallest depths. Both these techniques eliminate points farther then a certain distance. One could also employ weighted minimization to handle sensor noise, but as we wished to optimize processing time, dropping points is more efficient.

The results for the fixed threshold (Figure 4.12, left) showed that below 1.5 m, this method does not yield enough points to ensure registration. As the threshold increases from 1.5 m to 5 m, the performance and the processing time follow a similar curve, essentially monotonic. The reason is that the average depth of what is detected changes while the sensor moves, and setting a fixed threshold leads to a lack of points in some situations. On the contrary, using a percentage of points has a different behavior, as shown in Figure 4.12 (right). Between a ratio of 0.4 and 0.6, the performance is higher than using all the points (i.e. with a ratio of 1.0) while the time is divided by half. Indeed, keeping less than 40% of the points reduces chances to take advantage of important constraints and using all the points does not cut off any noise. Therefore, in further experiments we selected the second technique with a ratio of 0.4.

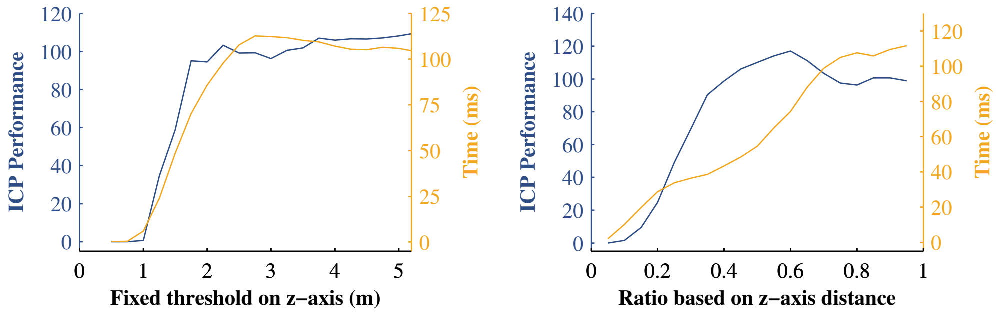


Figure 4.12: Performance and processing time for sensor-noise thresholds. Left: parameters based on fixed distances Right: parameters using quantiles. On both graphs, the ICP performances are in dark blue and the time in light yellow.

Subsampling

The second experiment evaluates how much we can subsample the point cloud without loosing too much performance. Again, we compared two subsampling techniques: (1) the random selection of points using a uniform distribution, and (2) the selection of only one point every n points (called fixed step technique). More complex subsampling techniques exist to compensate radial distribution of 3D scanners [Gingras et al. 2010] or to select points leading to a more constrained minimization [Gelfand et al. 2003], but these are too slow to cope with real time processing at 30 Hz.

We observed that processing time follows linearly the ratio of points used while performance follows an exponential convergence (Figure 4.13, left). The step technique results (Figure 4.13, right) showed an exponential reduction of the time while the general tendency of the performance is to reduce linearly. It is worth noting that parameters of the fixed step technique are discrete, which is shown using the filled dots on the curves. The performance of the subsampling step showed more jitters than the one of the random selection. We attributed this to artificial patterns in scans due to the fixed step nature of this technique.

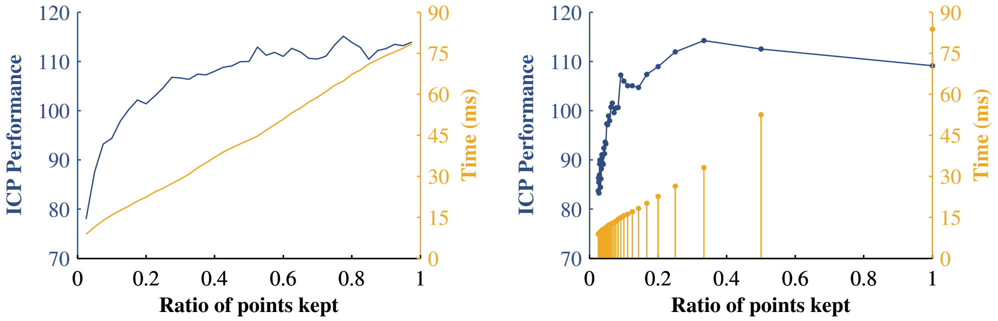


Figure 4.13: Performance and computing time for subsampling methods. Left: random selection. Right: fixed step based on n points skipped and expressed as ratio of point kept to ease comparison. On both graphs, the ICP performances are in dark blue and the time in light yellow.

We concluded that the random-subsampling technique gives us more control on the desired computation time and is less likely to produce artifacts in the resulting scans than the fixed step technique. Moreover, the comparison of computation time of both techniques in relation with the number of points kept and the extra computation time required for the random sampling showed that it does not augment the computational time significantly. Since there is no optimum for that parameter, we accepted the fact that going fast increases error and we thus selected a subsampling ratio of 0.3.

Nearest-Neighbor Approximation

This experiment stemmed from the observation that using an approximate NN-search leads to a faster ICP computation without affecting the error much [Nüchter et al. 2005; Zlot and Bosse 2009], when compared to an exact search. We implemented the NN-search using an approximate kd-tree as by Arya and Mount [1993] and varied the approximation factor ϵ.

Figure 4.14 (left) shows that when ϵ increases, both the computation time and the performance decreases, but the latter decreases slower than the former. Moreover, the time decreases rapidly to a minimum and then increases again. The reason is that while the number of points visited in the kd-tree decreases exponentially with ϵ, the number of iterations required by the ICP to converge increases linearly (Figure 4.14, right). Given those results, we selected $\epsilon = 3.3$, which is interestingly the same optimal value as reported briefly by Zlot and Bosse [2009].

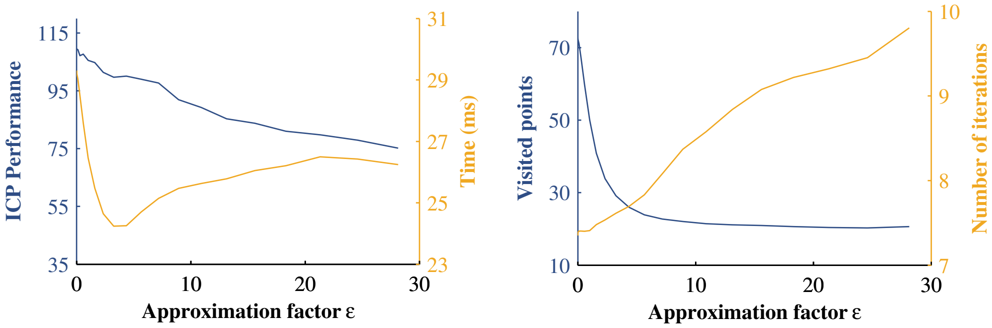


Figure 4.14: Performance and computation time for approximate search using a kd-tree. Left: ICP performances are in dark blue and convergence time in light yellow. Right: The average number of visited point per NN request expressed [10^3] are in dark blue and the number of iterations per ICP in light yellow.

Robustness Evaluation

Using the selected parameters from the previous section experiments, we compared the tracking error for different motion velocities, motion types, and environment complexities using the Kinect data sets (recall Figure 3.11).

Figure 4.15 presents the results of the tracker translation error directly in meters instead of the performance metric used in former experiments. Each of the subgraphs are a projection of all 27 data sets. For example, the bar graph for slow motion englobe all three types of motions (i.e., translation, rotation and free-fly) and three types of environment complexities. The error on translation for the three graphs is represented following a common log scale on the y-axis to highlight differences for low values. The box plots represent quantiles with the vertical red line being the median (A50), the blue box being the range [0.25, 0.75] and the dashed lines the range [0.05, 0.95].

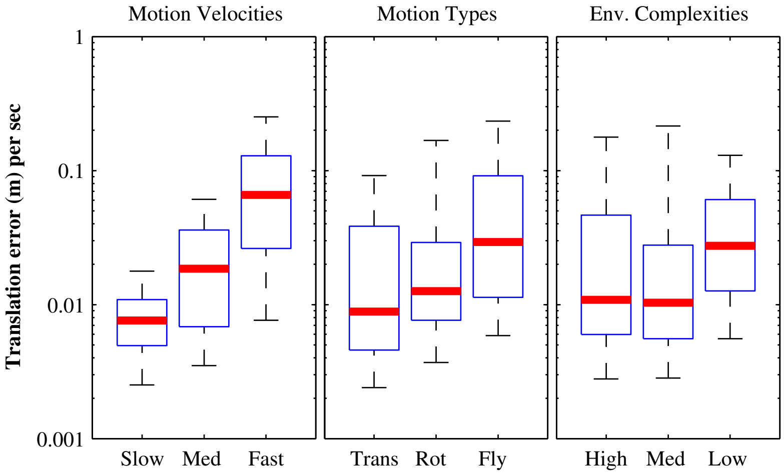


Figure 4.15: The error as function of motion velocities (slow, medium, fast), motion types (translation, rotation, free-fly), and environment complexities (high, medium, low).

The most important relation is the error that increases significantly as a function of the motion velocity, with the median being outside the first quartile for each velocity cluster. We also observed this effect with the success rate, which has a median value of 1.0 for the slow motion and going down to 0.68 for the fast motion (shown in Figure 4.16). Translation motions are easiest to register, followed by rotational movements and free-fly movements, where larger accelerations are present for both motion types. We noted that the low complexity environment is harder to register compared to high and medium complexity environments. The likely reason is that the low complexity environment contains very few planes and they are rarely all contained in the field of view of the Kinect, leading to some under-constrained dimensions.

In our experiments, the main factor influencing the registration speed was the number of points randomly subsampled. Since this processing time highly depends on the computer, we tested three different processors, keeping an increasing number of points. Note that the algorithm is not

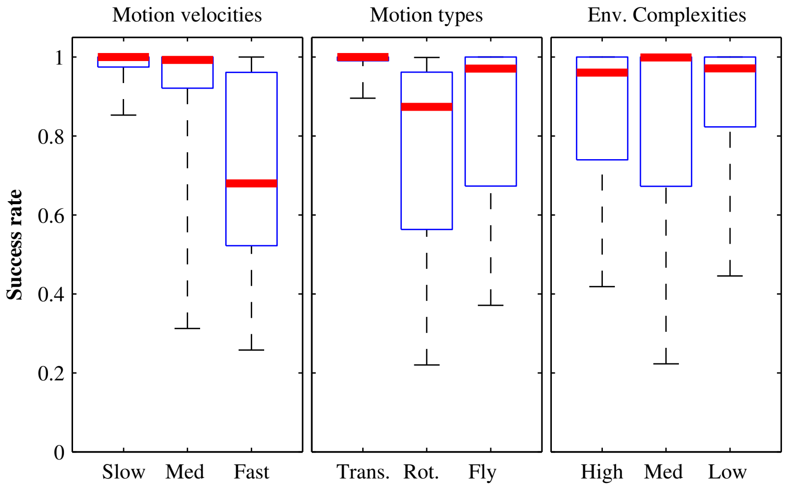


Figure 4.16: ICP success rate as function of motion velocities (slow, medium, fast), motion types (translation, rotation, free-fly), and environment complexities (high, medium, low).

multi-threaded and does not employ any GPU acceleration, which allows us to compare the performance with embedded systems. The three systems tested were: (1) a recent laptop with an Intel Core i7 Q 820 (1.73 GHz), (2) an old desktop PC with an Intel Xeon L5335 (2.0 GHz), and (3) an embedded system with an Intel Atom CPU Z530 (1.6 GHz).

Results in Figure 4.17 shows significant difference in the range of frequencies between the different systems. To ease the interpretation of the graph, a horizontal green line represents 30 Hz (i.e. the minimum frequency available for real-time operations using a Kinect) and a vertical green line represents the minimum acceptable number of points selected in the subsample experiment. Recent processors can process up to 3700 points at 30 Hz, while the Atom can run at most at 10 Hz, with the minimal number of points. Based on former experiments with quadcopters [Achtelik et al. 2011], a control loop needs to run between 5 and 10 Hz to cope with the dynamic of the system. Altogether, results showed that our tracker is usable on Unmanned Aerial Vehicles but those timing values excluded acquisition process. Recently, dual core Atom boards were release, which would give enough processing power for the acquisition and ICP.

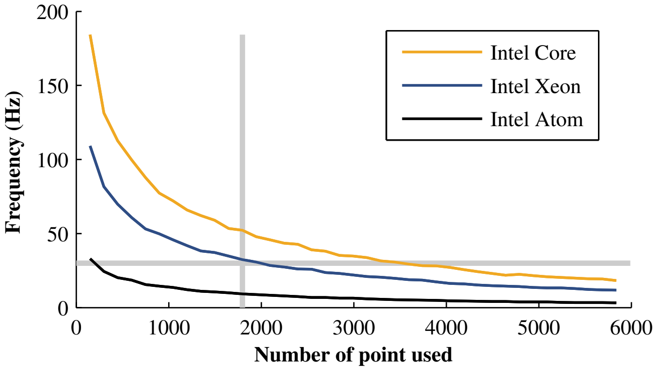


Figure 4.17: Comparison of time vs. ratio of points used for different hardwares. Vertical gray line correspond to critical number of points and the horizontal one to the real-time threshold.

4.3 Well-established ICP variants in Modern Applications

We wished to revisit two of the textbook ICP variants, using point-to-point [Besl and McKay 1992] and point-to-plane [Chen and Medioni 1991] distance metrics, both combined with the trimmed-ICP outlier rejection [Chetverikov et al. 2002]. We have chosen these because they are the most compared and researchers need to re-implement them every time. We hope to accelerate the comparison process for more modern solutions by providing complete evaluation results for those two baseline solutions. Henceforth, further comparisons can be made without of optimizing them to ensure fair analysis.

4.3.1 Variant Descriptions

Albeit simple, the variants depend on a certain number of parameters. We have fixed some and optimized others to allow for an efficient convergence of the algorithm. Table 4.4 shows the final values after optimization. We aimed at both minimizing the error and maximizing the time performance using the knowledge gathered while doing characterization of a fast tracker in Section 4.2.

Our ICP chain starts by sub-sampling both the reference and the reading point clouds. In the case of point-to-point, both point clouds are sub-sampled with uniform probability using the `RandomSampling` module. We explored the space of sub-sampling ratios using probabilities of keeping points in the range of {0.001, 0.01, 0.05, 0.1, 0.5, 1.0} for the reading and {0.001, 0.01, 0.05, 0.1, 1.0} for the reference. In the case of point-to-plane, because we wanted to extract the normals, we used the `SamplingSurfaceNormal`

	Step	Module	Description
Point-to-point	Data filtering of reference	`MinDist`	keep points beyond 1 m
		`RandomSampling`	random sub-sampling, keep 5 %
	Data filtering of reading	`MinDist`	keep points beyond 1 m
		`RandomSampling`	random sub-sampling, keep 5 %
	Data association	`KDTree`	kd-tree matching with ϵ of 3.16
	Outlier filtering	`TrimmedDist`	keep 75 % closest points
	Error minimization	`PointToPoint`	point-to-point
	Transformation checking	`Counter`	iteration count reached 150
		`Differential`	min. error below 1 cm and 0.001 rad
	Step	Module	Description
Point-to-plane	Data filtering of reference	`MinDist`	keep points beyond 1 m
		`SamplingSurfaceNormal`	sub-sampling 7× and normal extraction
	Data filtering of reading	`MinDist`	keep points beyond 1 m
		`RandomSampling`	random sub-sampling, keep 5 %
	Data association	`KDTree`	kd-tree matching with ϵ of 3.16
	Outlier filtering	`TrimmedDist`	keep 70 % closest points
	Error minimization	`PointToPlane`	point-to-plane
	Transformation checking	`Counter`	iteration count reached 150
		`Differential`	min. error below 1 cm and 0.001 rad

Table 4.4: Configurations of ICP chains for revisiting well-established ICP variants. *Top*: point-to-point. *Bottom*: point-to-plane.

module. We explored sub-sampling ration of one out of {5, 7, 10, 20, 100, 200}. For the reading, we used the same sub-sampling method as for point-to-point, looking for ratios of {0.001, 0.01, 0.05, 0.1, 0.5, 1}. After an exhaustive search, this optimization returned a reasonable ratios of 0.05 for both the reference and the reading for point-to-point, and a ratio of 0.05 for the reading with a threshold of 7 points for the reference for point-to-plane.

The matching step looks for the nearest neighbors of every point using a kd-tree. We used the `KDTree` module, which has three parameters: the number of nearest neighbors in the reference to associate to each point in the reading, an approximation factor ϵ allowing a maximum error of $1 + \epsilon$ between the returned nearest neighbor and the true nearest neighbor [Arya and Mount 1993] and a maximal distance beyond which neighbors are not considered any more. We chose a value of 3.16 for ϵ because, as shown in Section 4.2, this value led to the fastest registration. Indeed, with a smaller ϵ, nearest-neighbor queries take longer, and with a larger ϵ, more iterations are required until convergence because of the matching errors. Following the original implementation, we used only one neighbor and did not set any distance limit to the association.

We then rejected outliers whose distance is larger than a certain quantile. Using the `TrimmedDist` module, we explored keeping a ratio of {0.2 0.5 0.7 0.75 0.8 0.85 0.90 0.95 0.9999}. Based on this search, we decided to keep the 75% closest points for point-to-point and 70% for point-to-plane.

4.3.2 Evaluation against the *Laser Registration Data Sets*

We executed our protocol for both solutions leading to a total of 80,640 registrations (i.e. 2 solutions × 6 data sets × 35 paired scans × 3 types of perturbation × 64 perturbations). The overall translation results propose that point-to-plane (A50 = 0.76 m) is more accurate by 20 % than point-to-point (A50 = 0.97 m) solution. The advantage is reversed when looking at the difference between A95 and A50, which shows that point-to-point is more precise by 30 %. The same trend is observed for the rotation with the accuracy gain cranking to 40 % for point-to-plane while the precision advantage stays at 30 % for point-to-plane. For a deeper investigation, all results in Table 4.5 are subdivided into three categories: (1) data sets, (2) perturbation levels and (3) distance metrics. We can observe once more that most of the times the results of point-to-plane are better than point-to-point. Point-to-point error can however out-perform point-to-plane error for hard perturbations.

To explore the influence of the environment, Figure 4.18 compares the translation error combining all perturbations for each solution. Note that the A95 values for *ETH* exceed the graph, being 12.16 m for point-to-point and 16.87 m for point-to-plane. Focusing on A50 and A75, we see that the gain of point-to-plane over point-to-point is overcome in the data sets *Wood* and *Plain*. This observation proposes that the accuracy of each solution follows the level of structure found in each data set. When looking at the A95 statistics, point-to-plane is in all cases higher than point-to-point, meaning that point-to-plane does not guarantee better worst-case errors than point-to-point. It is worth noting that *ETH* consists of a long hallway with repetitive elements, which seems to drag down the A95 performance in translation while keeping reasonably low rotation errors (see Table 4.5). The data set *Plain* has an even higher deficiency in term of constraints than *ETH*, with only one major plane representing the ground. Even with this level of constraint, the registrations applied in *Plain* seem to diverge less than in *ETH* for hard conditions represented by A95 statistics.

			Apartment			*Stairs*			*ETH*			*Gazebo*			*Wood*			*Plain*		
			A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95
Translation	EP	Plane	0.06	0.47	2.11	0.09	1.17	3.49	0.10	0.44	6.06	0.11	0.38	2.08	0.25	1.55	4.75	0.42	1.54	4.15
		Point	0.13	0.54	1.54	0.35	1.29	2.57	0.47	2.23	6.86	0.28	0.60	1.71	0.39	1.48	4.21	0.51	1.46	3.09
	MP	Plane	0.20	1.04	2.98	0.61	2.08	4.64	0.60	4.06	16.3	0.28	0.96	3.51	1.25	2.92	6.62	1.30	2.58	5.58
		Point	0.46	1.03	2.32	0.94	1.86	3.38	1.92	4.29	11.2	0.49	1.13	3.18	1.19	2.52	5.15	1.21	2.17	3.76
	HP	Plane	1.35	2.18	3.66	2.05	3.28	5.50	4.18	8.55	19.6	1.87	3.33	6.95	2.79	4.52	7.86	2.35	4.13	8.85
		Point	1.29	1.99	3.24	1.81	2.78	4.75	3.84	7.06	14.8	1.58	2.79	4.57	2.32	3.73	6.82	2.02	3.14	6.33
Rotation	EP	Plane	0.02	0.20	1.14	0.02	0.31	1.58	0.01	0.02	0.61	0.02	0.08	0.48	0.05	0.34	0.95	0.07	0.20	0.60
		Point	0.07	0.25	0.97	0.12	0.39	1.22	0.05	0.22	0.83	0.04	0.17	0.41	0.09	0.29	0.77	0.09	0.20	0.44
	MP	Plane	0.08	0.47	1.80	0.16	1.08	2.09	0.01	0.25	2.91	0.04	0.35	0.97	0.31	0.78	1.53	0.19	0.38	0.99
		Point	0.20	0.61	1.49	0.33	0.78	1.63	0.14	0.59	1.82	0.15	0.35	0.80	0.32	0.69	1.22	0.20	0.37	0.77
	HP	Plane	1.01	1.72	2.95	1.48	1.91	2.94	1.31	2.09	3.11	0.58	1.31	2.88	1.05	1.56	2.53	0.50	1.09	3.05
		Point	1.04	1.60	2.53	1.10	1.64	2.53	0.97	1.73	3.05	0.58	1.20	2.59	0.97	1.44	2.35	0.46	0.99	2.09

Table 4.5: Overall view of the precision obtained with our two proposed baselines for different perturbations (easy (EP), medium (MP), hard (HP)). *Top*: Translation error [m]. *Bottom*: rotation error [rad]. Darker tones correspond to high error.

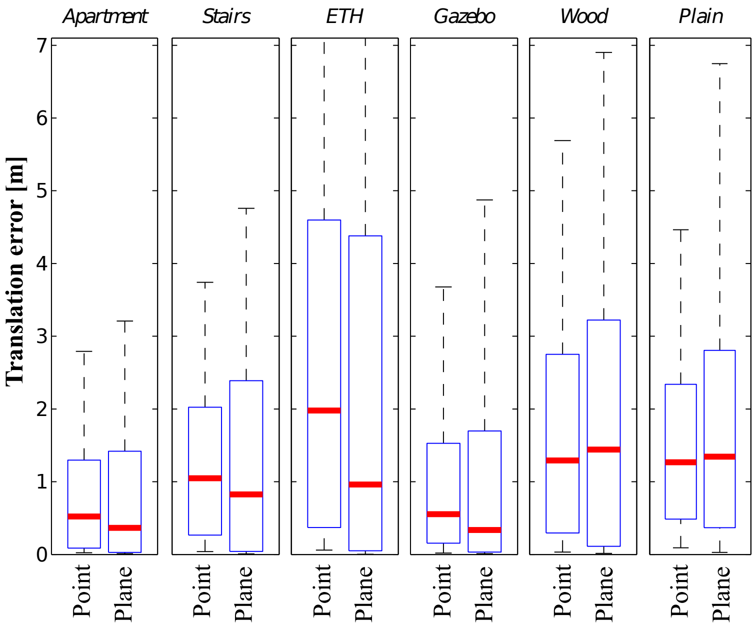


Figure 4.18: Comparison of point-to-plane and point-to-point performances for all perturbations and clustered environments. Thick red bars correspond to A50 (i.e. the median); the higher end of blue rectangles are A75 and the top end of dashed lines are A95.

Given that point-to-plane has a better overall performance, Figure 4.19 focuses exclusively on that solution and shows the cumulative probabilities of its translation error. Those curves are similar to precision-recall graphs in that the more top-left the curve the better the algorithm performs. The top plot emphasizes the influence of the environments given easy perturbations. This type of situation would happen for a mobile robot able to maintain low uncertainty on its localization between registrations. All of the environments keep their median error under 10 cm except *Wood* and *Plain*. Although considered a semi-structured environment, *Gazebo* keeps lower error, with *Apartment*, than the other environments. The bottom plot goes a bit deeper in the analysis by expending the results for *Apartment* to assess the influence of the perturbation levels. Each curve is associated with its initial perturbation level represented as a filled area. Ideally, all pairs of scans would have fewer residual errors after the registration leading to curves closer to zero than their associate perturbation level. One can observe that, for all perturbation types, roughly 25 % of the registrations still present worse translation than their initial perturbations. We believe the cause to be mainly the weak robustness of the solution against a range of different overlap ratios.

To demonstrate this low performance, Figure 4.20 shows the relation between the pre-computed overlap between scans and the translation errors for both solutions over all environments and all perturbation types. The statistics A50, A75 and A95 were extracted for each bin of paired scan sharing the same overlap, with the bin size being 0.08. Both solutions share the same *Outlier Filtering Module* tuned to handle 70 % and 75 % of outliers. This results in both solutions following the same trend leading to poor performance for low overlap values. The error reaches a median error larger than 2 m for a range of overlap from 0.30 to 0.38.

Finally, Figure 4.21 shows the cumulative probabilities of the time needed to converge for point-to-plane. The figure opposes structured environments (solid lines) to unstructured and semi-structured environments (dashed lines). It is interesting to note that in *Plain* the solutions converge rapidly but, based on Table 4.5, to a large translation error. This means that the observed errors were estimated to be below 1 cm and 0.001 rad (see the line Transformation checking in Table 4.4) leading to an early exit out of the iteration loop. For the overall performance between the two solutions, point-to-point is 80 % faster than point-to-plane with a median time of 1.45 s compared to 2.58 s respectively. This suggests that for point-to-plane, the extra time required to extract surface normal vectors is not compensated for by the saving on the number of iterations required to converge. All the results

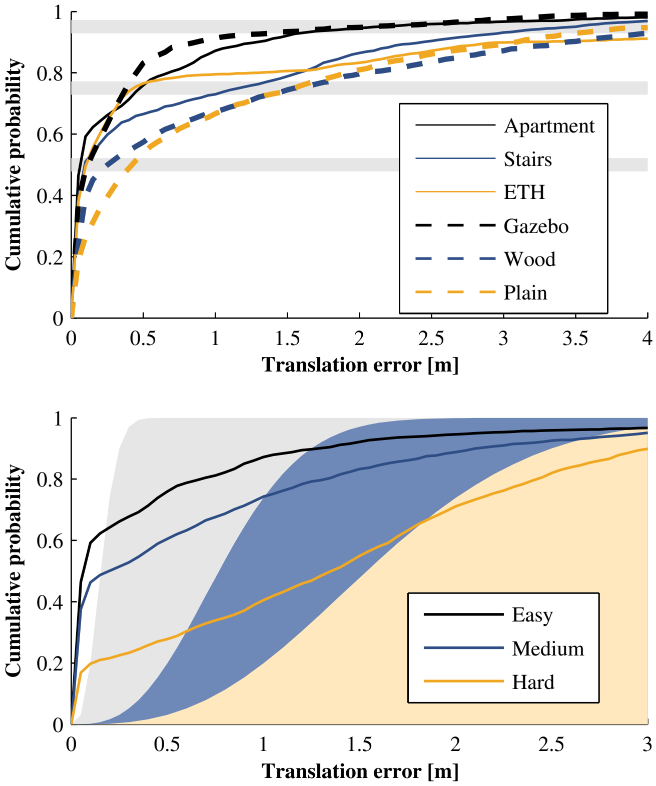


Figure 4.19: Cumulative probabilities of errors for point-to-plane ICP variant. *Top:* influence of environments given an easy perturbation level. The gray stripes correspond to the quantiles of interest, namely A50, A75 and A95. *Bottom:* influence of the three perturbation levels on the *Apartment* data set with the filled backgrounds correspond to the theoretical curves of initial perturbations.

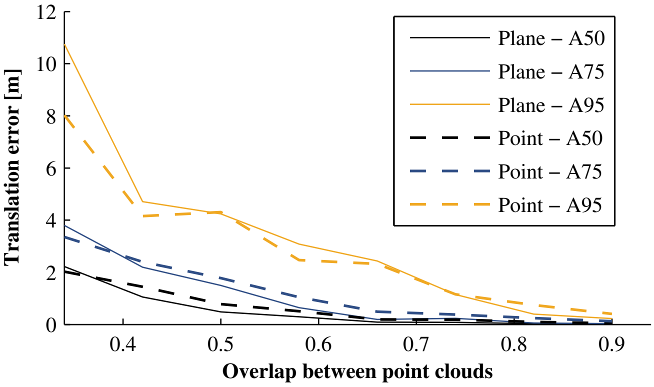


Figure 4.20: Correlation between the overlap of two scans and the translation error for point-to-plane over all environments and all perturbation types.

were obtained on a 2.2 GHz Intel Core i7-2675QM, using `libpointmatcher`.

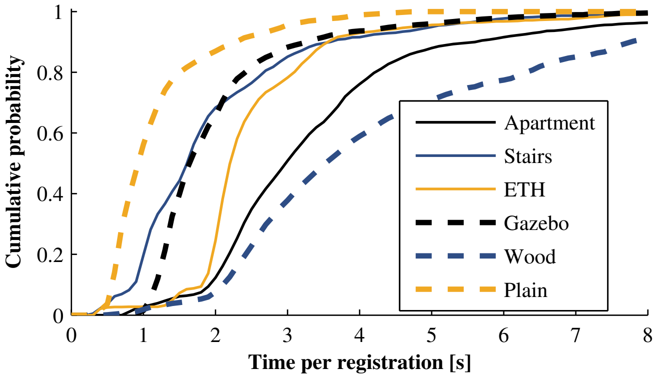


Figure 4.21: Cumulative probabilities of the time needed to converge for point-to-plane with easy perturbations. The solid lines represent structured environments while dashed lines represent unstructured and semi-structured environments.

We have sub-sampled the point clouds using a fixed reduction percentage leading to the use of approximately 10,000 points per scan. However, the different data sets have a different number of points per scan in average, for instance *Apartment* has twice as much as *Stairs*. It would be better to reduce the point clouds to a fixed number of points instead of a ratio to ensure more constant processing time given that the precision gain is very low for a larger number of points. As demonstrated in Figure 3.14, overlap between scans can largely vary depending on the motion of the robot and the environment configuration. One of the limitation of trimming outliers based on quartile is that this assumes a constant overlap of scans, which is hard to control with a mobile platform. In order to work around this limitation, it would be important to detect those places and react appropriately. For example, the robot could acquire scans more frequently or reduce its velocity at those places. Also, more flexible outlier-rejection algorithms need to be investigated to cope with the variability of the overlap.

The use of the A95 statistic might seem excessive, but it is important to note that it implies that one registration over 20 is beyond this value. In the robotics context, this is very significant and can be the difference between a stable system and a system that breaks its map every so often.

The point-to-plane solution can be stable for applications where: first, the environment type can be controlled to be highly structured; second, the overlap is kept high while the robot is moving and third, the state estimation used as initial pose for the registration remains within 10 cm and 10°. These types of conditions are usual for laboratory experiments but are unlikely to happen in real applications.

4.4 Summary

Through this chapter, we proposed a better characterization of key factors influencing registration algorithms. We had a deeper look at some popular depth sensors, addressed real-time registration issues and assessed the impact of potential environments on well established algorithms. All of those elements being selected with respect to robotic constraints with the aim to provided better intrinsic and extrinsic evaluations related to geometric registrations.

More precisely, we proposed two types of model for the random noise of the URG-04LX, the UTM-30LX and the LMS-151 scanning laser rangefinders. We believe that the methodology used provides more realistic models than in prior studies. Evaluation of surface reconstruction was also tackled, leading to the conclusion that a NN search radius should be larger than 5 cm to overcome the noise of the sensors. The implementation those models in `libpointmatcher` should help further solutions to better handle typical noise sources in their registration algorithms.

Based on this library, we designed and optimized a 3D pose tracker for dense depth cameras running at 30 Hz on standard laptop with thousands of points. As it does not use GPU acceleration, the tracker can also be run on embedded system (at 10 Hz on an Atom board). We proposed a sound performance evaluation using data sets recorded with a ground truth of millimeter precision. When it comes to optimization, it is very difficult to find a general solution to all problems using ICP. We can optimize a particular ICP implementation by identifying environmental characteristics and typical motions expected for a given application. One must also take into account sensor frequency, noise, and field of view to devise a robust registration strategy. Once a strategy allowing the ICP to converge properly is found, one can further optimize the speed given a minimal expected accuracy on the pose. The speed of ICP is linear with the number of points in the *reading* point cloud, but the pose error decreases exponentially. This means that, at some point, a solution using twice the computation time will only slightly decreases the pose error. One must carefully consider this trade-off, especially in time critical application.

From a robotic-application point of view, pose tracking in cluttered rooms, typically encountered in apartments or offices, is easier than tracking in corridors of public buildings or in places with few furnitures. To cope with this, one could adjust the speed of the robot as a function of the complexity of the environment. One should also limit the rotational velocity when the curvature of the sensor path is large. Looking at results found in the literature, those considerations are often controlled by the person realizing the experimentation without being explicitly stated and quantified. It is suggested that novel solution evaluations should highlight range of intrinsic and extrinsic parameters instead of presenting well controlled cases where the algorithm work properly.

As a proof of concept, we also applied the protocol presented in Section 3.2 to classic ICP solutions. The procedure we proposed relies on some specific data sets in order to have a common ground of comparison in the scientific community. We shown that even if the point-to-plane distance metric is in general superior to the point-to-point distance metric, it can be less precise for large disturbances of the initial alignments and also no more reliable in unstructured environments. This refreshes the observations from Rusinkiewicz and Levoy [2001] by using data sets closer to robotic applications. The performances of these baseline variants show a high variability and strongly display the need for improved ICP methods for natural, unstructured and information-deprived environments. This need opens the door for other researchers to challenge their novel solutions against our baselines.

However, as the sensor is the same across all data sets, we cannot measure its effect on the ICP performances. The sensor has nevertheless two important features, noise and field of view, that can have an influence on ICP. Indeed, sensors may have different noise levels and even noise profiles, and different ICP variants might cope better with some than others. Furthermore, the field of view and the point-density profile of the sensor inside its field of view can have a huge influence on the ICP performance as those characteristics govern the overlap and the possibility of multiple pairings between scans.

Concerning time, some applications require online matching of sensor data. In these cases, the time spent in ICP is a relevant criterion to compare variants. However, processing time is difficult to measure given that internal memory management, processor load and processor types are all relevant factors that cannot easily be compensated for and that can drastically change time measurements. On the other hand, theoretical complexity is not sufficient as different ICP variants will mostly have a comparable complexity but different constant factors. Having a single computer dedicated to running all the different ICP variants in the same condition would yield a general idea of the relative efficiency. But still, different ICP variants would scale differently for different practical cases. A comparison of the variants in the specific case of application is thus always pertinent. Our library can facilitate this comparison by highlighting only the relevant changes. Indeed, the efficiency of an implementation is an important factor of time performance that can bias the comparison of algorithms. Having a library in which only the modules to be compared change already significantly reduces this effect by maintaining a homogeneous environment for most data processing.

Chapter 5
Applications

Building a 3D map of a given environment is rarely the final goal pursued when using a mobile platform. The 3D representation of an environment often support other algorithms or complement other sensors. In this chapter, we present different platform prototypes to demonstrate the utility of geometric registration in different applications. Following the analyses detailed in Chapter 4, we here describe those applications by focussing on the environment, platform velocity, available localization information and type of sensors. For all applications, the same computation scheme was applied. We used a fairly simple processing pipeline based on the concept of metascans proposed by Wulf et al. [2008]. A sequence of 3D scans were streamed to a registration module, which employed (1) a set of filters directly on the input 3D scan in its local frame of coordinate (i.e., the origin being the center of the sensor), then (2) a standard ICP that registered the new input scan with a global map and finally (3) a concatenation of the newly registered scan with the global map, followed by a set of filters applied to that global map. All of those processing steps were made using the library presented in Section 3.3. Only text-based files with parameters were changed to achieved the presented reconstruction. No loop-closing detection nor error back propagation algorithms were used to post-process the resulting maps. Indeed, this three-step procedure could be considered as *laser odometry* (as compared to visual odometry) instead of a globally consistent solution. The goal was to present qualitative mapping results, evaluate for which applications local registration is sufficient and to discuss about the impact of a specific application on the registration solution used. We had the opportunity to cover multiple type of platforms moving on the ground, air and water, with different sizes and velocities.

The first application presented in Section 5.1 addresses a critical task for real-time registration: Search & Rescue. Field deployments on firemen training site were published at the *2012 International Conference on Field and Service Robotics* [Kruijff et al. 2012], while the results specific to registration were accepted for publication in the journal of *Autonomous Robots* [Pomerleau et al. 2013]. A second application relevant to registration is the automation of inspection as described in Section 5.2. The mapping capability of the platform used to demonstrate this task was first published at *1st International Conference on Applied Robotics for the Power Industry* [Tâche et al. 2010], and the extended results were issued in the *Journal of Field Robotics* [Tâche et al. 2011]. In same section, we also briefly introduce a path planning dealing with the particular mobility and mapping capabilities of the platform used for inspection. Those results were published in the *International Journal of Robotics Research* [Stumm et al. 2012]. The third application employed a novel autonomous vessel, which was described in *IEEE Robotics and Automation Magazine* [Hitz et al. 2012], to demonstrate shoreline monitoring (Section 5.3). Finally, autonomous-driving car is shortly addressed in Section 5.4. We end in Section 5.5 with some lessons learned based on the experimentation in those different conditions.

5.1 Use Case: Search and Rescue

Within the framework of the European project NIFTi (FP7-ICT-247870), novel solutions were assembled together and tested in order to support firemen and first responders in Search & Rescue missions. A first use of 3D maps is to help strategic deployment of responders in environment were humans can only intervene for a limited time. Those situations include nuclear incident, chemical spill, unstable supporting structures, excessive heat conditions, etc. When tele-operated, 3D maps can be used to enforce the user situation awareness, supporting critical decisions about risky platform motions. This type of application often has a limited communication range, leading to an increased need for autonomous behavior. More autonomy also means more onboard computations in case of communication breakdown. In such situations, onboard localization is essential to bring back the platform were the wireless communication can be reestablished.

Apart from increasing pressure on real-time solutions, Search & Rescue environments cover a large spectrum of possibilities. For example, deployments can happen in a well-structured nuclear plant, in a partially-collapsed building or outdoors in a case of a train chemical spill. In an advanced robot-human collaborative intervention, dynamic elements created by the other agents (e.g., firemen, other robots, etc.) acting on the field need to be considered in the registration solution. Moreover, dynamic elements at a global level (e.g., building collapsing during the exploration) can happen. In the case of relatively contained situations where human can hardly access a restricted intervention area, very little dynamic elements are expected (e.g., the Fukushima incident). In the cases presented in this section, applications were demonstrated with few dynamic elements and without the need to identify them.

The platform deployed on the field was called NiftiBot (Figure 5.1). The mobility was ensured by a set of two tracks having a passive suspension system facilating the crossing of uneven terrain. Moreover, two flippers per track allowed an active control of the platform orientation and offered an extended range for gap traversals. The mechanical configuration of the robot enabled it to climb slopes up to 45°, including stairs. The robot occupied a volume of 0.17 m^3 and weighted approximately 20 kg. The primary sensor used for registration was a 2D rotating Sick LMS-151 laser, with its rotation axis pointing toward the front of the robot. The aggregation of 2D scans used the motion information of the platform to reconstruct 3D scan at 0.35 Hz. A typical 3D scan contained 55,000 points. Two other sensors can be found on the platform, namely a GPS-aided IMU (X-sens MTI-G) and an omnidirectional camera (PointGrey Ladybug2). The velocity of the platform can be considered slow (0.3 m/s), especially during teleoperation where delicate motion are required. The pre-alignment information used as input for the registration was based on a Kalman filter fusing the IMU and the odometry information. The large error on motion estimates came from the vibration of the tracks, the large contact surface of the tracks on the ground, and the fact that the platform often collided with obstacles and can punctually slip. Therefore, smooth and continuous motion models can easily break, thus simple prediction models (e.g., constant velocity) can hardly be applied.

The configuration of the rotating laser produced a high density of points in front of the robot, which was desirable to predict collision, but not beneficial to the registration minimization. Thus, we forced the maximal density to 100 points per m^3 after having randomly subsampled the point cloud in order to finish the registration and the map maintenance within 2 s. We expected the error on pre-alignment of the 3D scans to be less than 0.5 m based on the velocity of the platform and the number of ICP per second that will be executed. So we used this value to limit the matching distance. We also removed paired points with an angle difference larger than 50° to avoid the reconstruction of both sides of walls to collapse when the

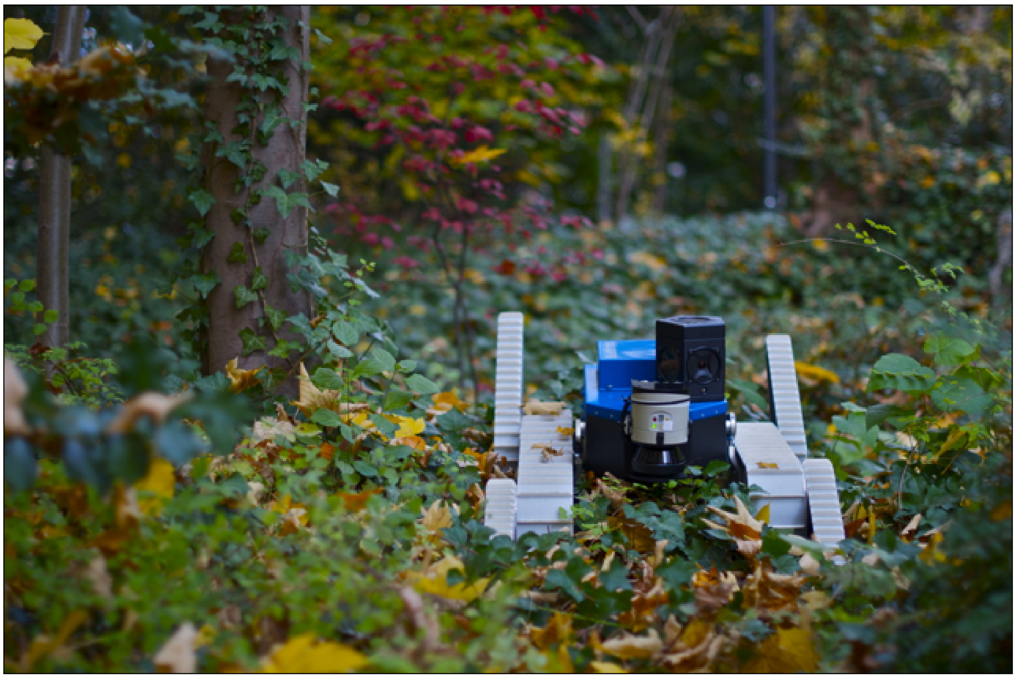


Figure 5.1: Photograph of NiftiBot, the main platform used for Search & Rescue demonstrations.

robot was exploring different rooms. As for the global map, we maintained a density of 100 points per m^3 every time a new input scan was merged in it. A maximum of 600,000 points were kept in memory to avoid degradation of the computation time performance when exploring a larger environment than expected. The complete list of modules used with their main parameters can be found in Table 5.1.

5.1.1 Indoor Preliminary Tests

To test the range of the platform motions and to demonstrate the need for 3D reconstructions, we ran an experiment in our laboratory. The robot started his path in an office located on the E-floor, and was driven down a staircase two floors below (C-floor, in the basement). The robot was controlled using a joystick by an operator following it throughout the path. The robot was then driven six floors up to the I-floor using the same staircase (Figure 5.2). In this application, the robot acquired scans from a stop-and-go strategy with a scan taken roughly every 2 m.

The complete map was processed onboard the robot respecting real-time

Table 5.1: Configurations of ICP chains for the NIFTi mapping applications.

	Step	*Module*	*Description*
Input	Read. filtering	`SimpleSensorNoise`	SickLMS
		`SamplingSurfaceNormal`	keep 80 %, surface normals based on 20 NN
		`ObservationDirection`	add vector pointing toward the laser
		`OrientNormals`	orient surface normals toward the obs. direction
		`MaxDensity`	subsample to keep point with density of 100 pts/m^3
Registration	Ref. filtering	-	processing from the rows Map
	Read. filtering	-	processing from the rows Input
	Data association	`KDTree`	kd-tree matching with 0.5 m max. distance, $\epsilon = 3.16$
	Outlier filtering	`TrimmedDist`	keep 80 % closest points
		`SurfaceNormal`	remove paired normals angle $> 50°$
	Error min.	`PointToPlane`	point-to-plane
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 40
		`Bound`	transformation fails beyond 5.0 m and 0.8 rad
Map	Ref. filtering	`SurfaceNormal`	Update normal and density, 20 NN, $\epsilon = 3.16$
		`MaxDensity`	subsample to keep point with density of 100 pts/m^3
		`MaxPointCount`	subsample 70 % if more than 600,000 points

computation constrain. Because the information of the past environment were fused in the global map while the robot went down the stairs, the drift in the localization was considerably reduced on the way up. This experiment comprised two critical moments: (1) when the robot moved out of the office and (2) when the robot entered the basement (C-floor). In both situations the overlap between the new scan and the global map went critically low. This information was known by the operator, so more scans were taken at those moments to avoid large deviations in the global map.

5.1.2 Rail Yard

On two occasions, the NIFTi platform was tested outdoors in a rail yard, with the permission of the *Depo kolejových vozidel Praha* (Prague Depot of Rail Vehicles, Czech Republic). In the first experiment, the robot was also driven in the yard by operators who were following the platform. The robot started its journey at one corner of a wagon, going along the wagon flank to the other corner and then, turned back to the starting position passing through the vegetation located on the other side of the wagon (Figure 5.3). Even if the path contained a loop, the precision of the registration was

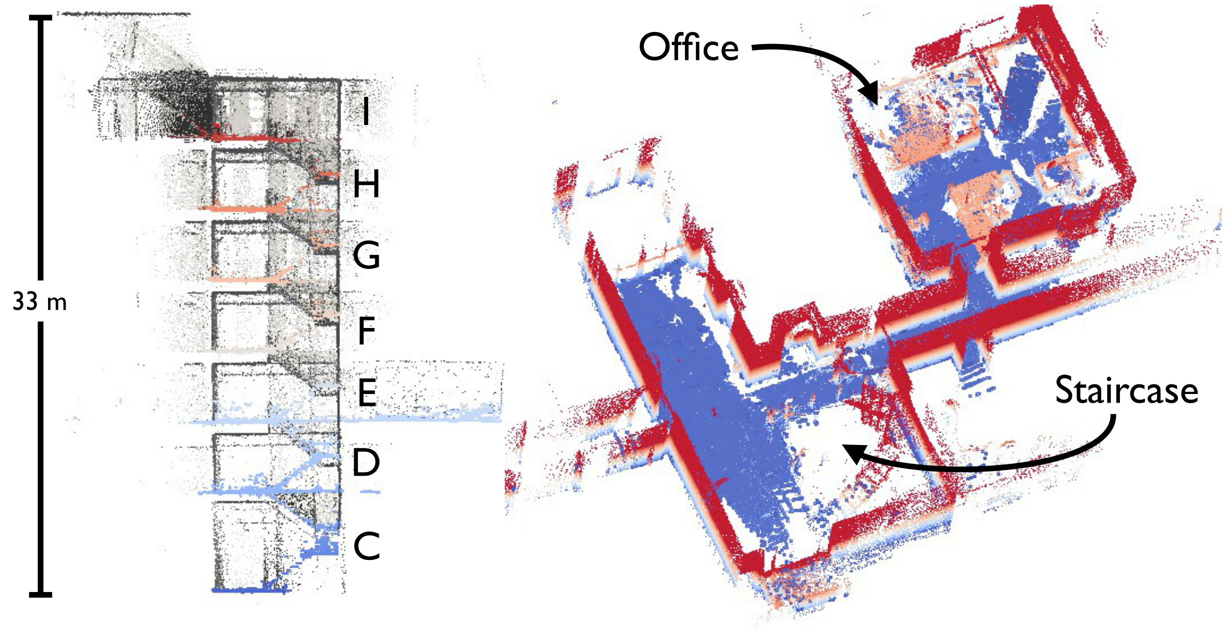


Figure 5.2: Mapping of a 7-floor staircase using a Search & Rescue robot. *Right*: Side view of the resulting map with the floors colored based on elevation. *Left*: Top view of the E-floor with the ceiling removed and the points colored based on elevation, red being higher, blue lower.

accurate enough to properly match the first scan with the ones recorded at the end.

In the second experiment, the robot was driven to explore inside of an old wagon where a person was standing still in the shadow to test in parallel the capability of the thermal camera. The robot then went out of the wagon, crossed dense vegetation, followed the side of a more modern wagon and stopped in front of it, where a second operator was captured in the global map.

For both experiments, the robot acquired scans with a stop-and-go strategy. The scans were gathered at uneven distances (up to 8 m apart) by operators without prior knowledge about critical situations. All the 3D scans were processed offline four times faster than the speed at which they were recorded. We thank people from the Czech Technical University (Prague), who are partners in the NIFTi consortium, for recording and generously providing the data.

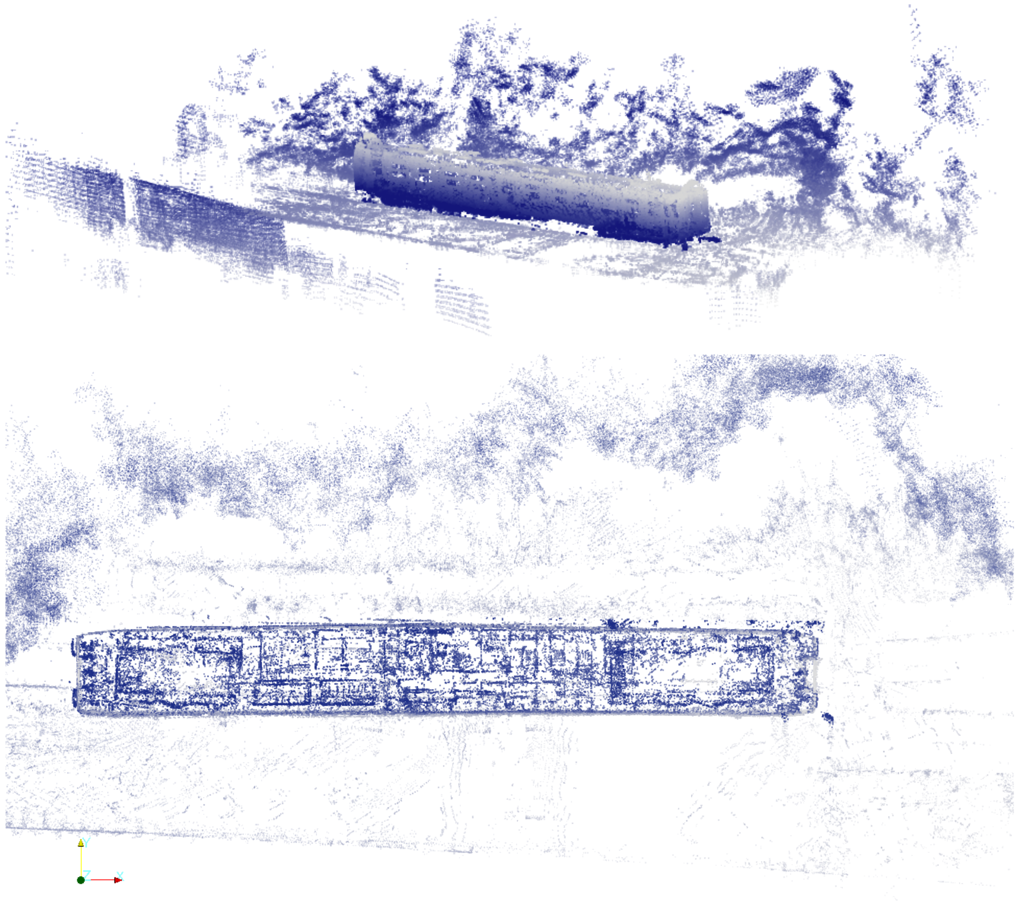


Figure 5.3: Deployment of the NiftiBot in a rail yard with a single wagon and dense vegetation. *Top*: Side view of a the reconstructed environment with the wagon in the middle and the vegetation behind. *Bottom*: Top view of the reconstructed environment. For both images, colors of the point clouds were chosen to ease the comprehension of the 3D-scene.

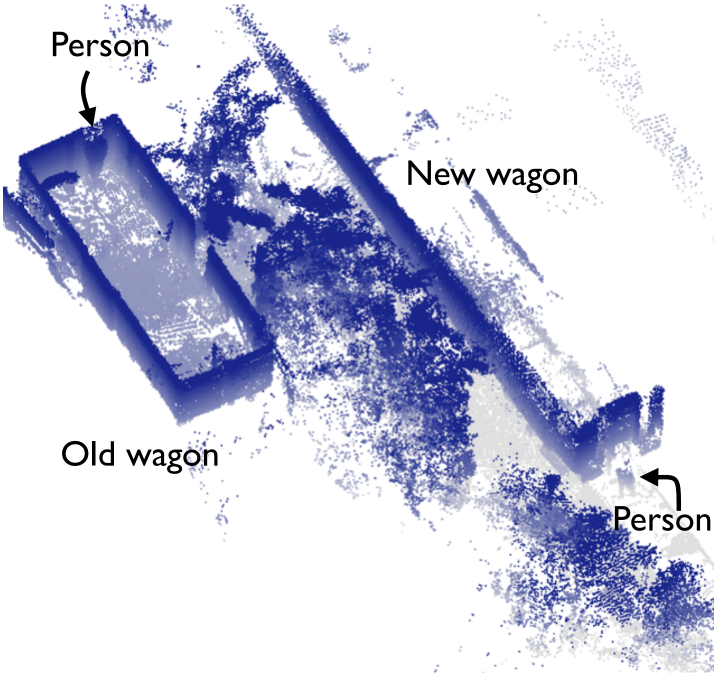


Figure 5.4: Bird view of the second experiment where the robot explored inside an old wagon and followed a more recent one.

5.1.3 Collapsed Church

In May 2012 a sequence of earthquakes hit the Emilia-Romagna region, Northern Italy, with a magnitude of 5.8. Three month later, NIFTi partners deployed the platform with the support of the *Vigili del Fuoco* (National Fire-watchers' Corps of Italy) and the *Beni Culturali* (Ministry of Culture of Italy) in Mirandola for a damage assessment mission. One of the visited sites was the Chiesa di San Francesco d'Assisi, in which 3D scans were recorded. The robot started outside the church, crossed a door and realized a straight line, navigating on the cluttered floor of the western gallery of the church (Figure 5.5). One can observe on the reconstruction the pillars and arches supporting the remaining roof of the church. The level of damage of the church was quite important, thus limiting the exploration possibilities of the platform, as depicted in Figure 5.6.

The platform was remotely operated from a control station situated outside the church and was able to continuously scan the environment while moving in the environment. Again, all the 3D scans were processed offline four times faster than the speed at which they were recorded. We thank our NIFTi collaborators from University La Sapienza (Rome) for recording and generously providing the data.

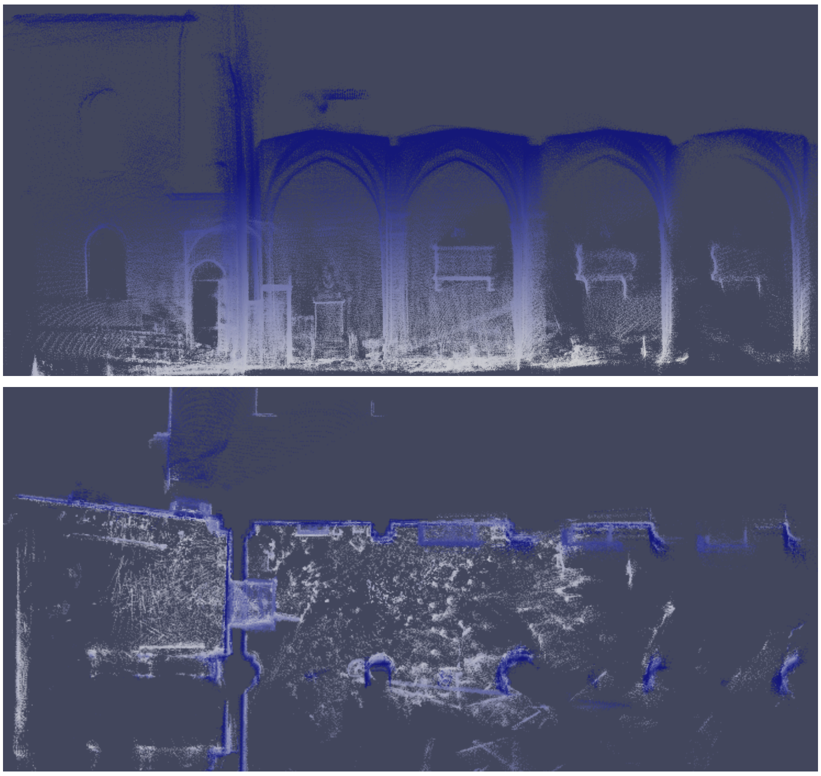


Figure 5.5: Reconstruction of the San Francesco d'Assisi Church with the color following elevation. *Top*: Side view of the reconstruction. *Bottom*: Top view of the church.

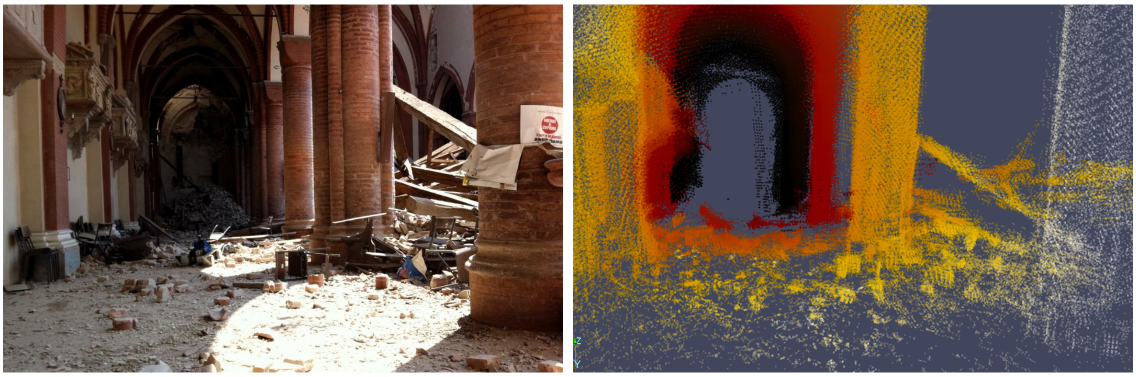


Figure 5.6: Comparison of the point cloud reconstruction with a photograph taken during exploration. *Left*: Photograph of the western gallery with the collapse roof on the right. *Right*: Front view of the reconstruction.

5.1.4 Collaborative Mapping

Within the framework of the European project *sFly* (FP7-ICT-231855), three micro-helicopters (AscTec Firefly) were deployed over a Search & Rescue training site in Zurich, Switzerland. The platform used is shown in Figure 5.7 flying over a collapsed concrete building. The three Fireflys were sent so that each one covered a pre-determined part of the environment and streamed back images to a control station. The collected images were used by the ETH Computer Vision and Geometry Group to reconstruct a 3D representation of the environment explored (Figure 5.8 - *Bottom left*).

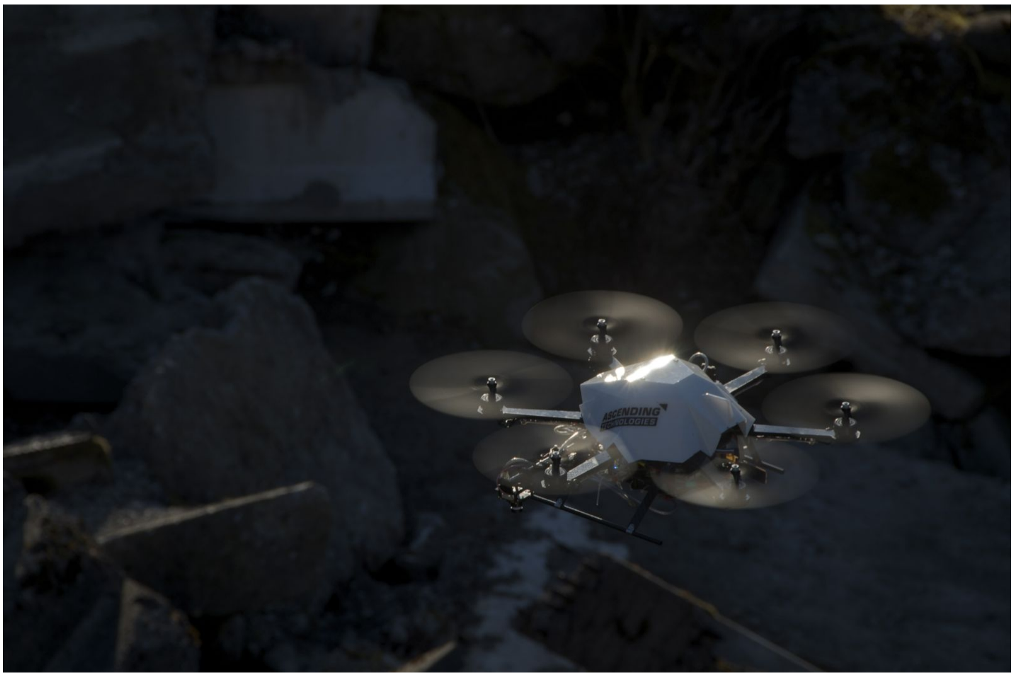


Figure 5.7: Photograph of one of the three AscTec Firefly used to map the environment in collaboration with the NiftiBot.

Another map was realized using a ground platform (i.e., NiftiBot). The robot was tele-operated on a road around the main collapsed building presented in Figure 5.8, for a path totaling 110 m long. The operator had a good prior knowledge of the environment before driving the robot around from a control station. The large road coupled with the awareness of the environment contributed to increase the velocity of the robot while exploring the area. The resulting map of the ground robot and the map of the Fireflys were then fused using a standalone ICP implementation taken from our registration library `libpointmatcher`. Both map were having roughly

300,000 points and were registered using the configuration of Table 5.2. The final map is depicted at the bottom right of Figure 5.8.

Table 5.2: Configurations of ICP for the collaborative mapping applications.

	Step	*Module*	*Description*
Registration	Ref. filtering	`SamplingSurfaceNormal`	surface normals based on 50 NN, keep 15 %
	Read. filtering	`RandomSampling`	keep randomly 15 % of the points
	Data association	`KDTree`	kd-tree matching with 1.0 m max. distance, k-NN = 5
	Outlier filtering	`TrimmedDist`	keep 80 % closest points
	Error min.	`PointToPlane`	point-to-plane
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 100

At that time, the Kalman filter used to fused the odometry with the IMU was not well tuned and bias in the estimation were inducing drift on yaw estimates. Roughly, a constant drift of 5°/s was estimated visually. The scans were gathered while the ground platform was moving, which generated a larger localization error than prior experiments. A total of four runs were recorded with the ground platform: (1) continuous scanning, turning clockwise around the main building; (2) continuous scanning, counterclockwise; (3) stop-and-go scanning, clockwise; and (4) stop-and-go scanning, counterclockwise. Two experiments out of four closed the loop with a negligible error at the closing point. Surprisingly, the successful runs were the ones turning counterclockwise, contradicting our first intuition that stop-and-go scanning would be more accurate. This experiment highlighted the importance of correcting IMU drift with an external registration algorithm and also showed that the robot could scan while moving. We selected the resulting representation of the second run to fuse both sources of information (i.e., laser and camera) in a common 3D reconstruction.

5.1.5 Artor - Autonomous Rough Terrain Outdoor Robot

An other platform was developed in parallel to the NiftiBot by the Autonomous Systems Lab (ASL). Modifications were done over a LandShark from Black-I Robotics, in collaboration with RUAG and Armasuisse. The aim of the project was to develop techniques for reliable autonomous navigation of a wheeled robot in rough, outdoor terrain.

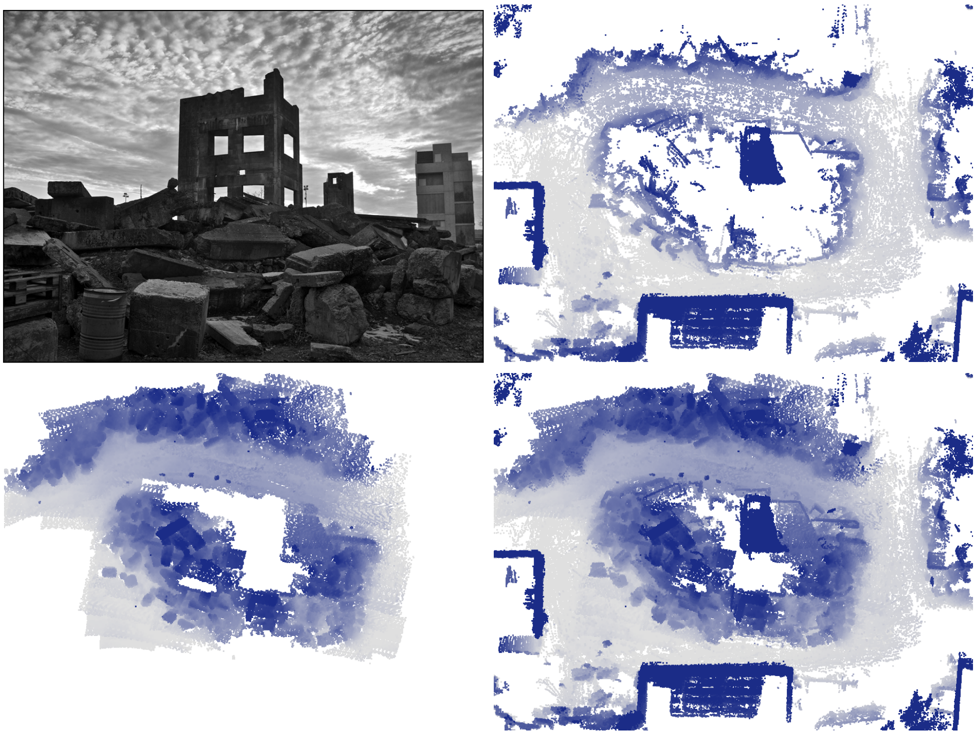


Figure 5.8: Resulting maps of the Zurich firefighter training site. *Top right*: Photograph of the training site with a partially collapsed tower in the middle. *Top left*: Top view of the reconstruction realized with the data from the ground robot. *Bottom left*: Top view of the reconstruction realized with the data from the three Fireflys. *Bottom right*: Top view of the combined map. Note that the color correspond the elevation: blue is low, red is high.

The robot, named Artor, was much larger than NiftiBot, with a volume of 0.96 m^3 and an approximate weight of 350 kg (Figure 5.9). Three wheels on each side of the robot gave the same traction as tracked vehicles, while simplifying the maintenance of the locomotion system. Odometry suffered from the same large rotation error problem as NiftiBot because of the unknown friction between the ground and the wheels. The robot can drive at a maximum speed of 4.5 m/s but is usually driven at around 1.2 m/s. The motion of the platform can be smooth on the pavement but in off-road situation, the motion can be more rough, and at high speed the orientation can change critically fast. Odometry computation can lead to large error that is mainly caused by the high friction of the wheels on the ground. Given the early development stage of the platform, only the wheel odometry was available as prior information for the registration module. The main sensor

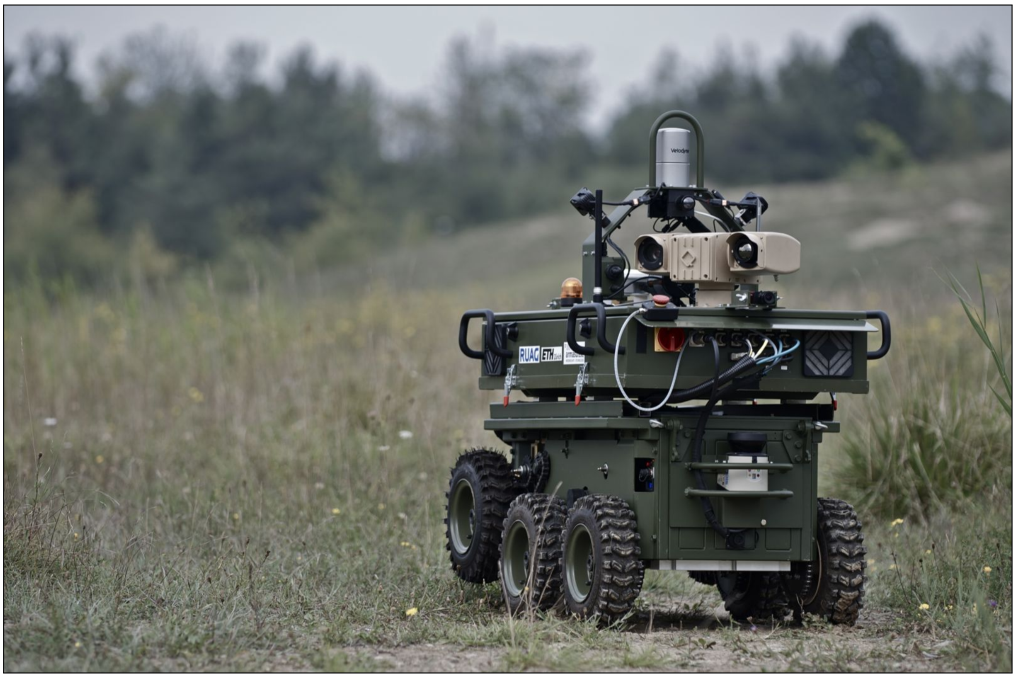


Figure 5.9: Photograph of Artor, a Search & Rescue robot specialized for outdoor applications.

used in this experiment is the Velodyne HDL-32E, which produces around 50,000 points at 5 Hz. Other sensors present on the platform were: two Sick LMS-151 (front and rear), high-resolution zoom camera, thermal camera and a GPS-aided IMU.

The critical element for real-time processing is the amount of points that needs to pass at high rate through the registration module. The tuning evolved from the parameters and filters selected for NiftiBot with the aim of increasing the registration speed for Artor. We first randomly removed 85 % of the points to ensure a stronger data reduction. We also kept a lower density of points because the platform usually covered larger areas than the NifitBot during a typical deployment. The complete list of modules used with their main parameters are listed in Table 5.3.

The Artor robot was driven over different types of terrain in a specialized testing facility in Wachtberg, Germany. The path of the platform was 340 m long following a rectangle shape and no noticeable errors were found at the loop closure. The robot realized the same loop for a second time using the global representation of the first loop without any problems. An overlay of the 3D map top view with an orthogonal projection of an aerial image is

Table 5.3: Configurations of ICP chains for the Artor mapping applications.

	Step	*Module*	*Description*
Input	Read. filtering	`RandomSampling`	keep randomly 15 % of the points
		`SurfaceNormal`	surface normals based on 20 NN, $\epsilon = 3.16$
		`ObservationDirection`	add vector pointing toward the laser
		`OrientNormals`	orient surface normals toward the obs. direction
		`MaxDensity`	subsample to keep point with density of 50 pts/m^3
Registration	Ref. filtering	-	processing from the rows Map
		`MaxDist`	keep points within a radius of 70 m
	Read. filtering	-	processing from the rows Input
		`RandomSampling`	keep randomly 25 % of the points
	Data association	`KDTree`	kd-tree matching with 5.0 m max. distance, $\epsilon = 3.16$
	Outlier filtering	`TrimmedDist`	keep 90 % closest points
		`SurfaceNormal`	remove paired normals angle $> 90°$
	Error min.	`PointToPlane`	point-to-plane
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 40
		`Bound`	transformation fails beyond 5.0 m and 0.8 rad
Map	Ref. filtering	`SurfaceNormal`	Update normal and density, 20 NN, $\epsilon = 3.16$
		`MaxDensity`	subsample to keep point with density of 10 pts/m^3
		`MaxPointCount`	subsample 70 % if more than 600,000 points

provided in Figure 5.10 - *Top right*.

The data were processed offline at the same rate as the recorded one. The operator was driving the platform around to test the mobility capability of the robot, without explicitly considering any registration limitations. On an open terrain, the solution proposed can manage a global and dense representation of the environment, even with the high turn rate of the robot. This might be more challenging for a solution based on visual odometry to avoid drift in those conditions. We thank Philipp Krüsi (ASL) for recording and generously providing the data.

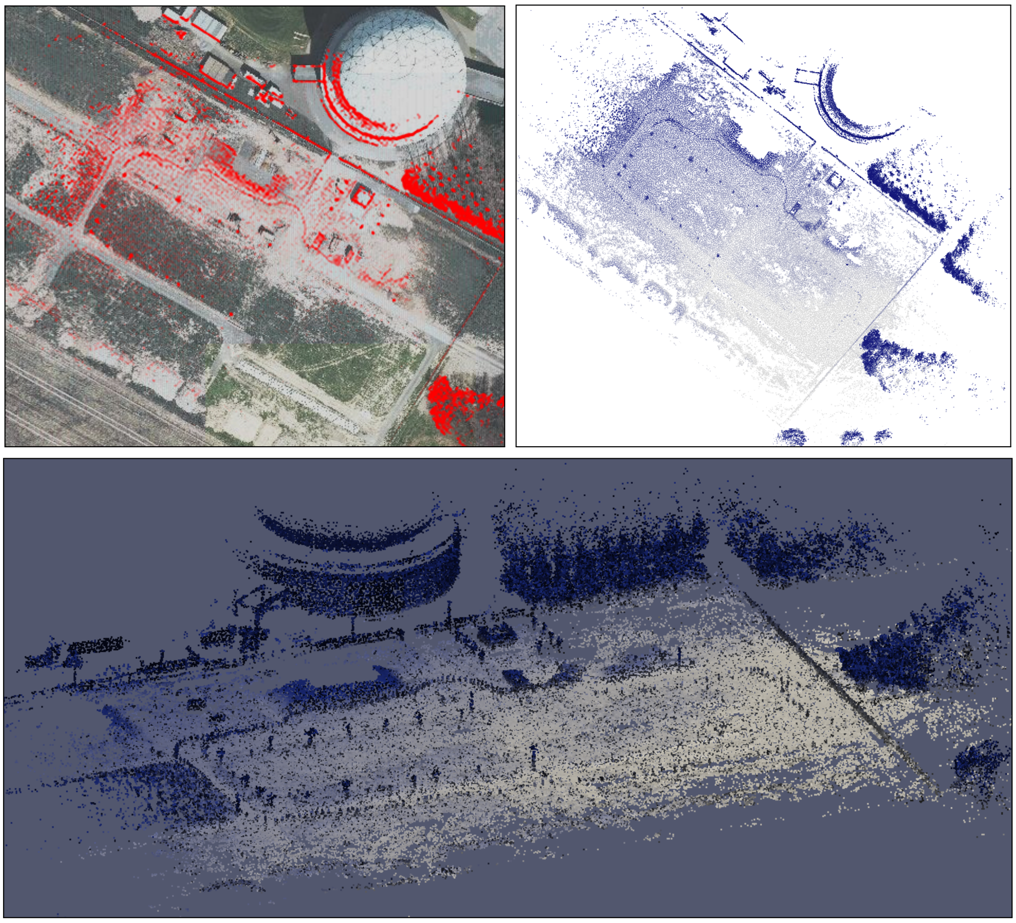


Figure 5.10: Reconstruction of the testing facilities. *Top right*: Overlay of the 3D map with an aerial view. *Top left*: Top view of the reconstruction. *Bottom*: Bird view of the reconstruction. The color is based on the elevation of the points, light gray being low, dark blue being high. The aerial image was provided by *Bundesamt für Kartographie und Geodäsie*, Frankfurt, Germany - `http://www.bkg.bund.de`

5.2 Use Case: Power Plant Inspection

In collaboration with Alstom Inspection Robotics (AIR), prototypes were developed for the inspection and maintenance of industrial plants. Some inspection tasks need to move inspection tools in environments that are difficult to access by human due to dimensional, temperature or air quality constrains. The use of mobile systems for inspection can not only deal with those constrains, but also can reduce the time and costs of inspections. This would, for example, allow for the inspection of critical pieces of equipment on location, without the need to dismantle any structures. Similarly, the installation of scaffolding around a structure becomes unnecessary, thus saving inspection time. The typical environments encountered during inspection procedures are confined spaces (indoors) with well structured, static surfaces.

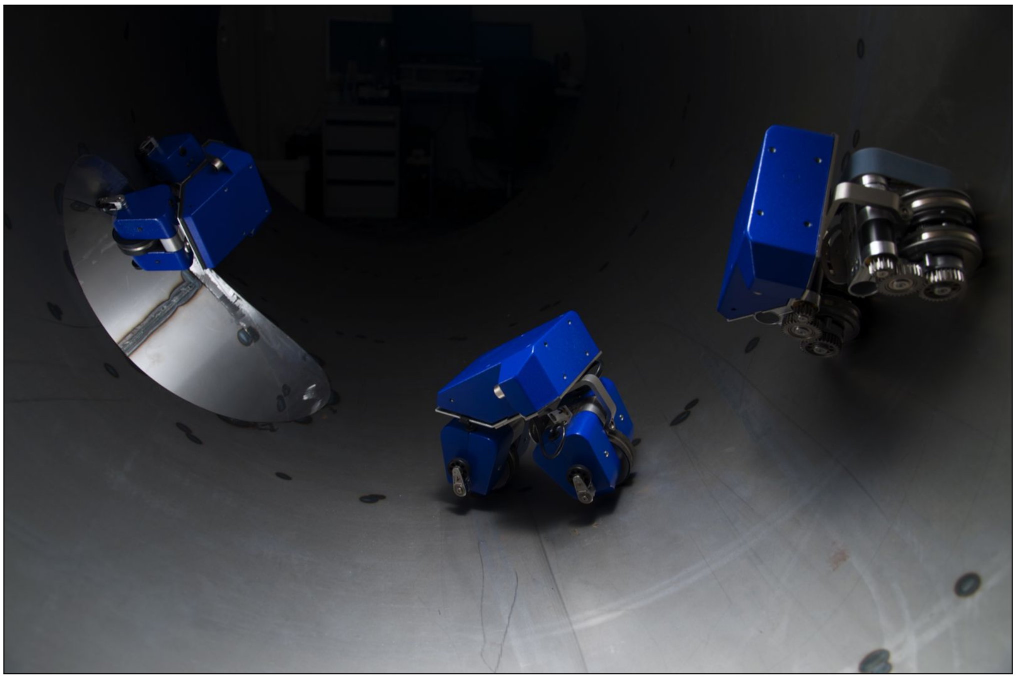


Figure 5.11: Three prototypes of chest inspection robots without the sensors, in a mock-up of a steam chest.

For the specific task of steam chest inspection, a robot was developed with high mobility capabilities and a compact size [Tâche et al. 2009]. The robot, named Magnebike, moved around a metallic, cylindric environment by using its two magnetic wheels positioned in the same configuration as a bike (Figure 5.11). Those specialized wheels coupled with small lever arms allowed the platform to move up-side-down, pass 90° edges and navigate in high curvature tube. A considerable amount of effort were invested in reducing the size of the platform, which ultimately gave a small volume of 0.006 m^3 and a weight of 0.34 kg. The platform was the slowest presented in this chapter, with a maximal velocity value of only 0.045 m/s. The robot gathered high-resolution scans (340,000 points) with a refresh time of 50 s. The 3D scans were assembled from a 2D Hokuyo URG-04LX. Pre-alignment of the scans were ensured by wheel odometry, which displays virtually no slip because of the magnetic force hold the wheels on the surfaces. An IMU was used in conjunction with the odometry to cope with the 3D nature of the motion [Tâche et al. 2011]. The main sources of pre-alignment errors were: (1) motions perpendicular to the gravity vector that were not observable with the sensor used, and (2) the rapid cumulation of errors by the low-cost IMU used on the slow-moving robot.

5.2.1 Steam Chest Reconstruction

During inspection, the robot is intended to be tethered for safety reason, which solved the communication problem between the embedded computer and a faster system used as control station. Because the system must not damage the inspected structure in any case, it traveled slowly in the environment, reducing the pressure on the real-time requirement for the registration. However, the operator might not have a visual contact with the robot at all times during inspection. Therefore, the map resolution must be high enough to detect obstacles and holes during remote operations. The number of points produced by the scanner was way beyond what was required for a proper registration. To reduce rapidly this number of points, we randomly removed 90 % of the point as soon as the scan were recorded. We used a maximal density of 20000 points per m^3 to cope with the small size of inspected environments. For the registration, we did not use any pre-alignment to test the worst case scenario (i.e., when the rotation is not observable by the IMU). This forced us to extend the maximal matching distance to 0.5 m. Given that scans are taken by an operator at a fix and short interval, we used an outlier ratio of 80 %. The complete list of modules used with their main parameters are listed in Table 5.4.

To test the mapping capability of the platform, a real steam chest was made available by AIR. This part was actually removed from a power plant for reparation purpose. Multiple inspection runs were executed, each run starting from one of the seven entry points (Figure 5.12). We only

Table 5.4: Configurations of ICP chains for the Magnebike mapping applications.

	Step	*Module*	*Description*
Input	Read. filtering	`RandomSampling`	Keep randomly 10 % of the points
		`SamplingSurfaceNormal`	surface normals based on 20 NN, $\epsilon = 3.16$, keep 80 %
		`ObservationDirection`	add vector pointing toward the laser
		`OrientNormals`	orient surface normals toward the obs. direction
		`MaxDensity`	subsample to keep point with density of 20000 pts/ m^3
Registration	Ref. filtering	-	processing from the rows Map
	Read. filtering	-	processing from the rows Input
	Data association	`KDTree`	kd-tree matching with 0.5 m max. distance, $\epsilon = 3.16$
	Outlier filtering	`TrimmedDist`	keep 80 % closest points
		`SurfaceNormal`	remove paired normals angle $> 50°$
	Error min.	`PointToPlane`	point-to-plane
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 100
		`Bound`	transformation fails beyond 5.0 m and 0.8 rad
Map	Ref. filtering	`SurfaceNormal`	Update normal and density, 20 NN, $\epsilon = 3.16$
		`MaxDensity`	subsample to keep point with density of 20000 pts/ m^3
		`MaxPointCount`	subsample 70 % if more than 600,000 points

present here the results from the longest path since it covered the entire environment. The robot started on one side of the steam chest, situated on the left of Figure 5.12. Each 3D scan was taken on a stop-and-go strategy at every 0.1 m. The total path covered a distance of 5.8 m for a total of 59 scans. All runs were registered offline approximately 10 times faster than the rate at which they were recorded. We thank Fabien Tâche and Andreas Breitenmoser (ASL) for providing the recorded point clouds.

5.2.2 Support for 3D Path Planning

When aiming at increasing autonomy of a mobile platform, path planning rapidly becomes an essential tool. The unusual motion capabilities of the Magnebike posed challenges for traditional path planning algorithms that are mainly applied on flat 2D environments. Even 3D path planner working on elevation map can not be used because of the tubular shape of the potential navigation surfaces. More advanced techniques plan on 3D mesh with the assumption that there is a low level of noise corrupting the representation. In the collaborative work of Stumm et al. [2012], we proposed a navigation solution that takes into consideration the limitations of 3D mapping to

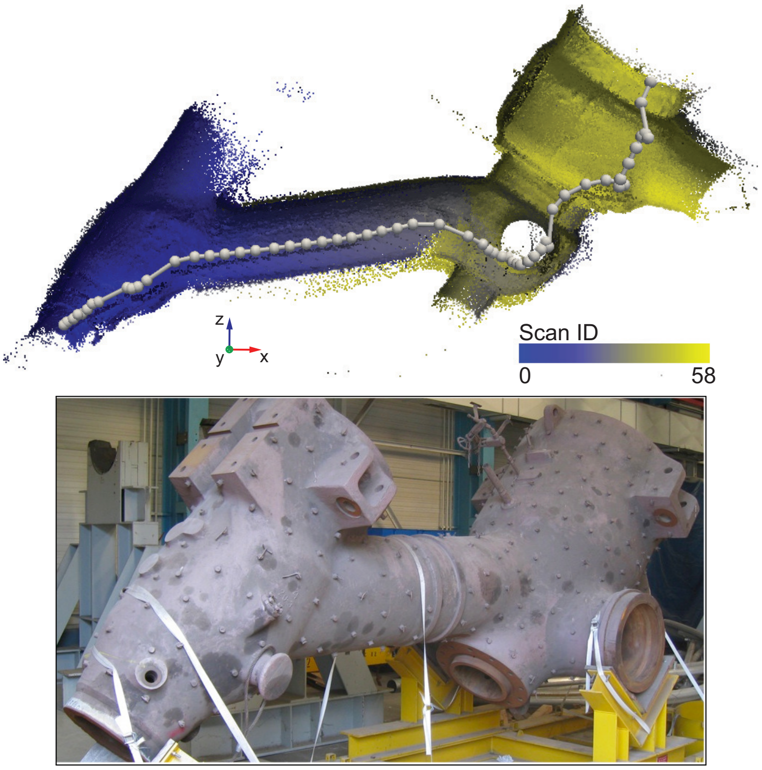


Figure 5.12: Deployment results of Magnebike in a real steam chest. *Top*: Cut view of the reconstructed environment. The white line correspond to the path of the robot, with the sphere being the positions where the robot stopped to take a 3D scan. The colors of the map follow the discreet time (from 0 to 58) at which the scans where taken. *Bottom*: Steam chest removed for maintenance.

enable the planning of a 2D path embedded in a 3D space. We only briefly introduce the results here to illustrate an application to 3D registration.

The environment structure was modeled from noisy point clouds that employed the tensor voting framework proposed by Medioni et al. [2000]. Tensor voting propagates structural information from points within a point cloud in order to estimate the saliency and orientation of the surfaces or curves in the environment. This framework also permits the detection of edge orientation, which constitutes important information in order to attack them at 90° (limitation of the Magnebike). The final representation outputted by the tensor voting provided a smooth interpolation of surfaces. This rendered possible the use of a specialized graph-based planner that established connectivities between robot states. The resulting sparse-graph structure eliminated the need to generate an explicit surface mesh, yet allowed for an efficient planning of paths along the surfaces, while remaining feasible and safe for the robot to traverse.

Different experiments in different simulated environments were realized to test the viability of the proposed navigation solution. Figure 5.13 shows one of those results, where the robot was asked to plan a path from one

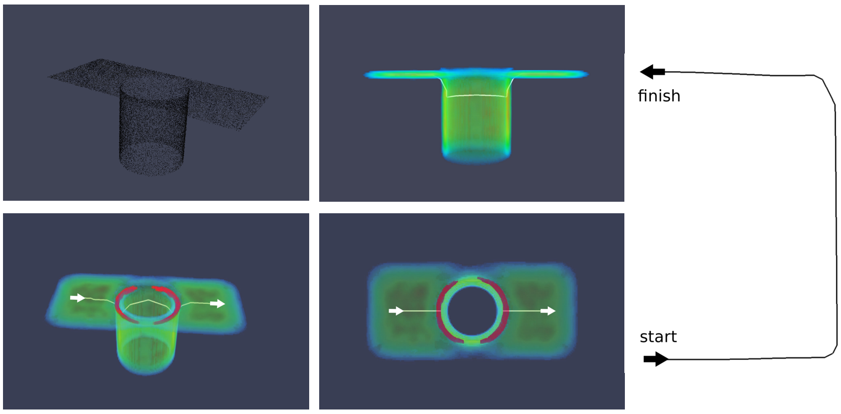


Figure 5.13: Proof of concept of the path planning solution on simulated points. *Top left*: Original point cloud. *Bottom left*: Perspective view of the reconstruction. *Middle*: Side and top projection of the reconstruction. *Right*: 2D path used for the motion controller. Note: 3D-planned path is in white and the edge detected as obstacles are in red in the reconstruction views.

side of a hole to the other. The figure presents the inputted point cloud with the surface saliency and the detected edges with the planned path. From the point of view of the robot, the resulting trajectory that need to be done moves forward in a straight line, turns left by 90°, continues again on a straight line, turns left a second time and continues ahead to the finish point. Although simple in 2D, the path required complex maneuvers in 3D, and dealt with surfaces and edges constrains.

Multiple paths were also planned on real 3D reconstruction. We reused the environment presented in Figure 5.12 and applied tensor voting to it, leading to the continuous representation at the top of Figure 5.14. Then, a starting point (S) was determined on the left side, from which safe paths were automatically computed to successfully reach five of the remaining exit points (1, 2, 3, 4, 5) out of seven. The solution is now being extended to platforms like NiftiBot, capable to move in 3D but limited by the gravity.

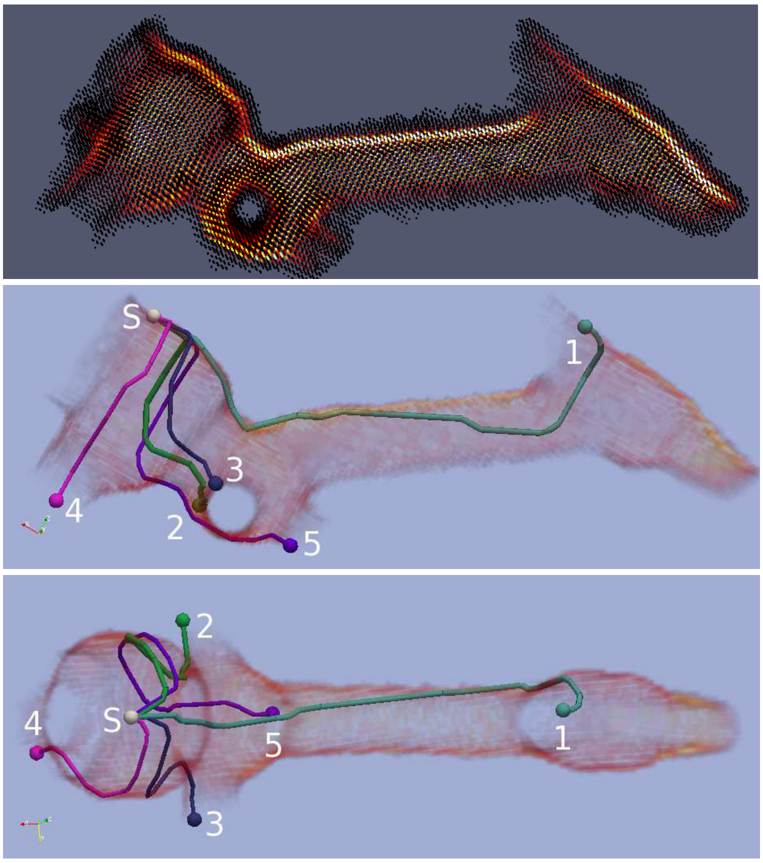


Figure 5.14: Results of the path planning algorithm in a real steam chest environment. *Top*: Cut view of the reconstructed environment with light yellow being an high probability of a surface. *Middle*: Side view of the five paths leading to different exit points. *Bottom*: Top view of the same five paths.

5.3 Use Case: Shoreline Monitoring

In order to support environmental monitoring of freshwater bodies, an autonomous surface vessel was developed in the ASL in collaboration with the Limnological Station of the University of Zurich. Although the vessel was initially developed to deploy biologic sensors in water (see [Hitz et al. 2012] for details on this application), a 3D laser was installed on its top to complement the analysis of the ecosystem with geological information. For example, 3D mapping of the shoreline could help to determine the volume of organic material (leaves) falling in a lake, accurately identify inflows of water, quantify coastal erosion, etc. The observation of coastal erosion using rangefinder laser is already an active field in geology [Mitasova et al. 2009], but it relies mainly on airborne surveys. However, this survey method can not provide a good viewpoint of a cliff, and the costs are quite high. The use of a boat as carrier is comparatively a low-cost method that can give better vantage points in certain situations. Beyond the geological applications, localization on the shore with centimeters precision can increase the autonomy of the system by allowing it to navigate to the sampling point from a parked position in a confined area, such as a boat house, or from its docking recharge station. One of the requirement of such application is to have long range for measurements, given that shores imply shallow water, which poses a limit on how close the boat can be without touching the bottom. The type of outdoor environments expected when surveying a water body can vary from structured to unstructured, depending on the intensity of the recreational use by the local people. For example scanned elements can be dams, bridges, houses, beaches, rocky shores, sparse to dense vegetation, etc. Except for other boats, most of the environment is expected to be static with potential for seasonal changes (global motion) monitoring.

The platform, named Lizhbeth, was deployed several times in Lake Zurich (Figure 5.16) and once in the alpine Lake Cadagno, both located in Switzerland. It had a volume of 6.75 m^3 and weighted approximately 120 kg. The motion of the robot was ensured by two electrical propellers positioned in the custom-built hulls of the catamaran. This gave differential drive motion capability to the platform, allowing it to turn on spot. The typical velocity of the robot is 0.7 m/s when surveying away from the shore. The main sensor used for 3D reconstruction was a Velodyne HDL-32E, which was configured to produce 45,000 points at 1.6 Hz. A single-beam underwater sonar was used to produce bathymetric maps. The localization sensors included an IMU, a magnetic compass and a GPS. The GPS was mainly used for offshore navigation because its precision of 5 m made it dangerous for nearshore navigation. The odometry can hardly be computed based on the motor inputs because of the high inertia of the boat in water, and the unknown wind-driven surface currents. Among the main sources of localization perturbations are the waves that may change rapidly the platform orientation, which can be evaluated by the gravity vector measured by the IMU. The smooth motion of the platform rendered difficult to extract reliable translation information without adding any registration algorithms. Nevertheless, a predictive model implying smooth 2D translations on the xy-plane can be used to pre-align scans.

While keeping the constraints of Lizhbeth in mind, we ran preliminary mapping experiments using the Velodyne installed on a small watercraft (7 m long). The substitute boat was a monohull and thus, was considered less stable on water than Lizhbeth, which is a two-hull vessel. As no external sensors were available, the full solution was tuned to not rely on any pre-alignment of the scans. The input filters applied ensured that the

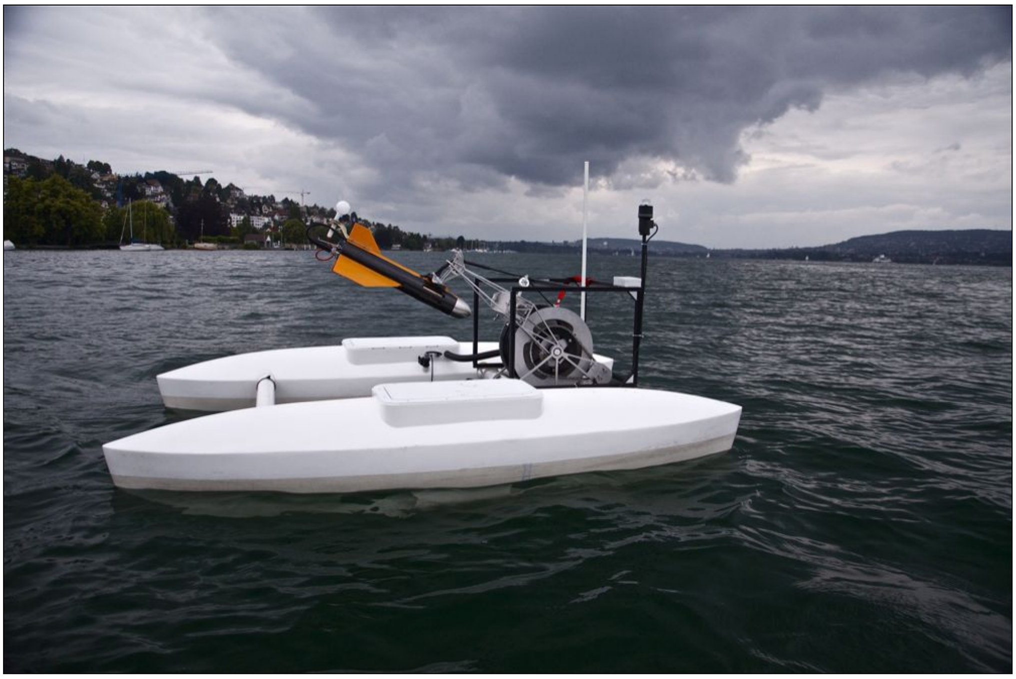


Figure 5.15: The autonomous surface vessel, named Lizhbeth, during one of its survey environment: pre-alpine Lake Zurich, Switzerland.

watercraft was removed from the scans, and a fast random subsampling reduced the number of points to ensure registration at every 0.6 s. The watercraft recorded scans while sailing, and its movements depended on water motions. Since waves induced fast changes in the watercraft's orientation, the matching of the scans needed to be fast enough to keep the error on the initial orientation small. The size of the survey area was expected to be large, so a low density of points were forced. When the laser hits the water it is usually not reflected back to the sensor, as opposed to solid ground. This has the advantage that a path can be planned on the space free of measurements instead of following a surface as presented in Section 5.2.2. Unfortunately, some waves can be detected by the laser because of their variable surface orientations. To reduce wave-reflectance effect, we applied a strict shadow point filter that removes 3D points that display an angle difference larger than 17° between surface normals and the direction of observation. The complete list of modules used with their main parameters are listed in Table 5.5.

The experiment executed with the watercraft was recorded on Lake Zurich, in front of the Limnological Station, the typical operating area of

Table 5.5: Configurations of ICP chains for the Lizhbeth mapping applications.

	Step	*Module*	*Description*
Input	Read. filtering	`BoundingBox`	remove points in a box of 7×7×2 m
		`RandomSampling`	keep randomly 80 % of the points
		`SurfaceNormal`	surface normals based on 20 NN, $\epsilon = 3.16$
		`ObservationDirection`	add vector pointing toward the laser
		`OrientNormals`	orient surface normals toward the obs. direction
		`MaxDensity`	subsample to keep point with density of 50 pts/m^3
		`Shadow`	remove points with angle between surface normals and observation direction $> 17°$
Registration	Ref. filtering	-	processing from the rows Map
		`MaxDist`	keep points within a radius of 70 m
	Read. filtering	-	processing from the rows Input
		`RandomSampling`	keep randomly 25 % of the points
	Data association	`KDTree`	kd-tree matching with 5.0 m max. distance, $\epsilon = 3.16$
	Outlier filtering	`TrimmedDist`	keep 90 % closest points
		`SurfaceNormal`	remove paired normals angle $> 90°$
	Error min.	`PointToPlane`	point-to-plane
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 40
		`Bound`	transformation fails beyond 5.0 m and 0.8 rad
Map	Ref. filtering	`SurfaceNormal`	Update normal and density, 20 NN, $\epsilon = 3.16$
		`MaxDensity`	subsample to keep point with density of 50 pts/m^3
		`MaxPointCount`	subsample 70 % if more than 600,000 points

Lizhbeth. The boat started away from the shore moving towards a harbor where multiple boats were parked side by side. This starting position is located on the lower left corner of Figure 5.16. The boat first passed between the harbor and boats anchored on buoys and turned right to continue between the anchored boats and the shore. The boat sailed parallel to the shore up to a boat house situated within an artificial small canal leading to the entrance of the warehouse. On the reconstructed environment, one can notice the noise around the anchored boats that was caused by their movements during the experiment, especially around the white one, at the bottom left corner. Also seen in that corner are the noisy light gray points that were generated by the reflection of the laser on the waves. The final map covered an area of 280 by 130 m without displaying any major defects.

The final solution must take into consideration that elements located offshore can have multiple possible positions. By keeping the global map updated at every uses, those multiple positions will be retained in the map,

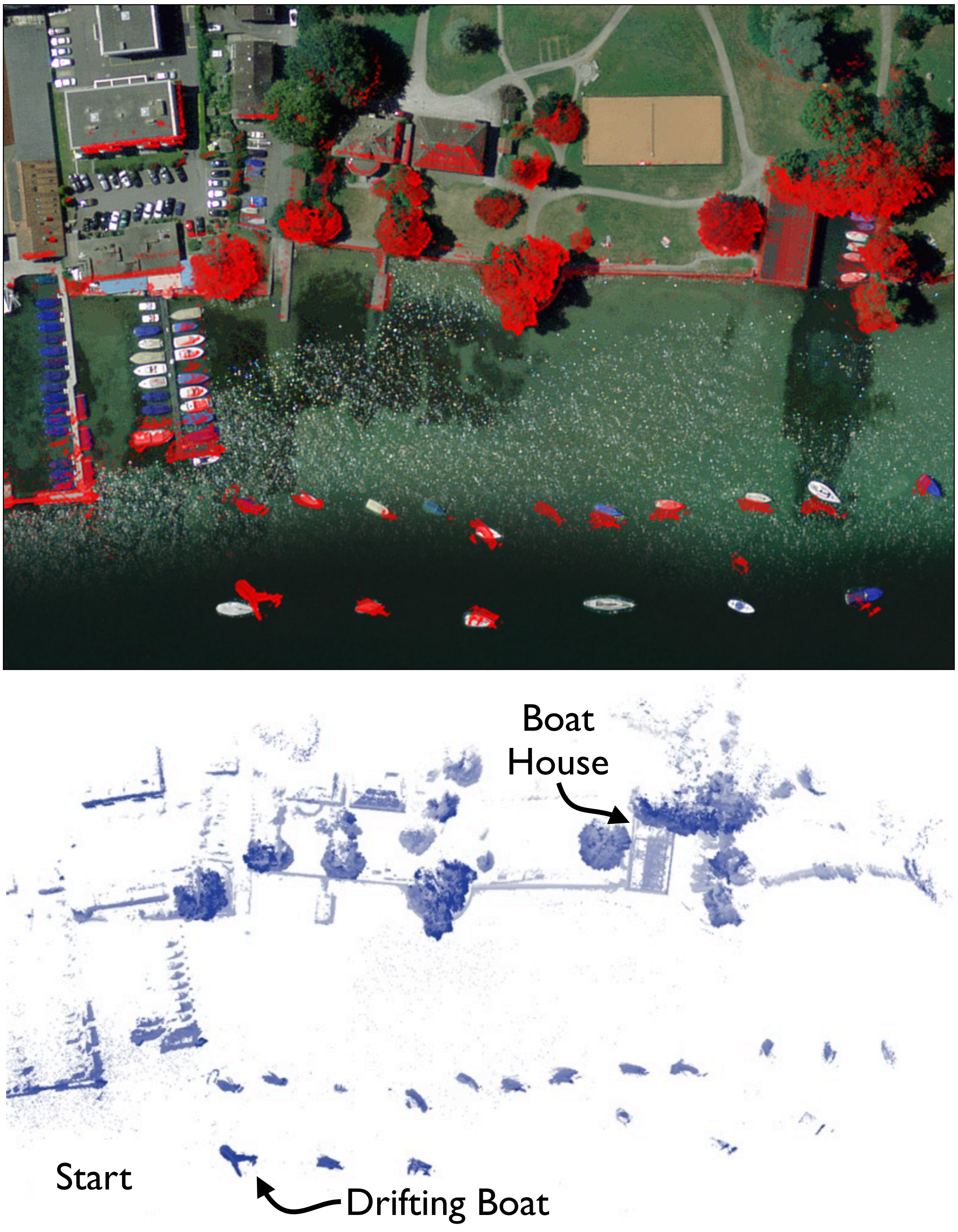


Figure 5.16: Reconstruction of the shoreline from a boat. *Top*: Overlay of the 3D map with an aerial view. *Bottom*: Top view of the 3D map with point colors based on elevation, light gray being low and dark blue being high. The orthogonal projection of the aerial image was provided by the Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000

thus reducing the chances of large drifts in cases where, for example, only the boats on the buoys were scanned. We thank Gregory Hitz (ASL) for its help gathering the scans.

5.4 Use Case: Autonomous Driving

The recent expansion of the autonomous driving field have been pushed forwards by car companies that teamed up with research partners, like Volkswagen in the EU project V-Charge, or by large company like Google hiring roboticists to develop new car prototypes. The American state of Nevada even officially delivered its first license for driverless car in May 2012. Although photometric registration is more attractive from the industry point of view, due to the potential low cost of cameras, geometric dense maps can support other activities. For example, road inspection is a tedious task that is mainly realized visually by operators driving on the roads. Large-scale road construction sites can also profit from a fast monitoring of the work progress. Applications usually target urban environments, which are mostly structured (e.g., road, buildings) or semi-structured when the vegetation is more prominent. The environment is predominantly static, but a large part of the field of view can be occupied by other cars such as in dense traffic situations.

The SmartTer (Figure 5.17) was a modified version of a Smart Fortwo. The car was developed by the ASL and served in 2006 has a technological demonstrator in the European SPARC project realized in collaboration with Daimler Chrysler. The Smart is one of the most compact car with a volume of 6.38 m^3 and a weight of 730 kg. Two Sick LMS-291 laser rangefinders were mounted on a vertical rotating axis, each of them providing 14,000 points every second. Motion compensations were applied to the 3D scans to cope with the high speed of the vehicle (15 km). Other sensors included navigation laser, omnidirectional camera, monocular camera, GPS and IMU. The overall motion of the vehicle is expected to be smooth, with a strong assumption of translation on the xy-plane. For a deeper description of the vehicle, we refer the reader to the publication of Lamon et al. [2006].

The solution selected for this application had to deal with a large scanned volume and noises caused by the velocity of the vehicle. We drastically reduced the number of inputted points by keeping a maximal density of 0.5 points per m^3. For the registration, we cut points beyond the maximal reach of the sensor to reduce the NN searching space. The pre-alignment of the scans was fairly accurate, but we kept a maximum matching distance of 1.5 m

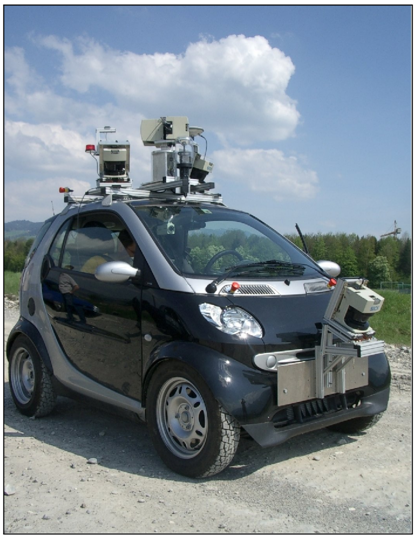


Figure 5.17: The autonomous car, named SmartTer, with its suite of five Sick LMS-291 laser rangefinders. Larger wheels were installed to allow off-road driving.

to recover from loop-closing error. Many large trees along the road were having their surface normal wrongly estimated due to their unstructured nature, which increased the point-to-plane alignment error. With that in mind, we used a point-to-point error metric for the minimization, as opposed to other solutions presented in this chapter. A denser global representation was maintained to ensure for more stability of the registration, especially on the ground where the density of a single scan was dropping rapidly.

Table 5.6: Configurations of ICP chains for the SmartTer mapping applications.

	Step	*Module*	*Description*
Input	Read. filtering	`SurfaceNormal`	surface normals based on 20 NN, $\epsilon = 3.33$
		`MaxDensity`	subsample to keep point with density of 5 pts/m^3
Registration	Ref. filtering	-	processing from the rows Map
		`MaxDist`	keep points within a radius of 40 m
	Read. filtering	-	processing from the rows Input
	Data association	`KDTree`	kd-tree with 1.5 m max. distance, $\epsilon = 0$, k-NN $= 3$
	Outlier filtering	`TrimmedDist`	keep 70 % closest points
	Error min.	`PointToPoint`	point-to-point
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 100
		`Bound`	transformation fails beyond 5.0 m and 0.8 rad
Map	Ref. filtering	`SurfaceNormal`	Update normal and density, 20 NN, $\epsilon = 3.16$
		`MaxDensity`	subsample to keep point with density of 10 pts/m^3
		`MaxPointCount`	subsample 70 % if more than 900,000 points

The complete ICP solution ran at 3 Hz, which permits the registration to run in real-time with a bit of margin. As showed in Figure 5.19, the SmartTer started and ended its path in a street situated at the top left corner of the aerial view. During the drive, a total of four loops were realized: two loops were made counterclockwise around the Eidgenössische Technische Hochschule (ETH) main building (Hauptgebäude), and two other loops clockwise around the Hospital (Universitätsspital Zürich). Even with the low density of point used, a reasonable amount of details could be preserved, as depicted in Figure 5.19 where even wires powering trams are visible over the street junction at the bottom of the image. At the first closing point, an alignment error of 4.5 m on the z-axis was recovered by the registration after a path of 660 m. The second closing point took place on a longer distance (920 m), with an error on the z-axis of 14.3 m, which error was also recovered by the registration a bit later, so the second loop could be done with the same representation without any problem. It is interesting to note that, at those two closing points, errors on z-axis were the most predominant of the 6 DoF possible. When looking at each scan separately, we observed that the ground area displaying a usable density of points had a radius of 20 m around the car. Also, the distance between two scans was often 8 m, going as high as 17 m when accelerating. Urban environment are often referred as canyon-shaped, which means that the sides are well constrained by the buildings, but lacking points on the ground can lead to drift on pitch angle or/and on z-axis. The fast motion of the car coupled with the short range measurements gave little overlap to stabilize the elevation or the pitch angle and the error cumulated during each loop. When the car was passing again in streets previously explored, the global map was reused successfully. This application results are the largest presented in this chapter, with a total path length of 3.8 km. We thank Rudolph Triebel and Luciano Spinello (ASL) for recording and preparing the scans presented in this section.

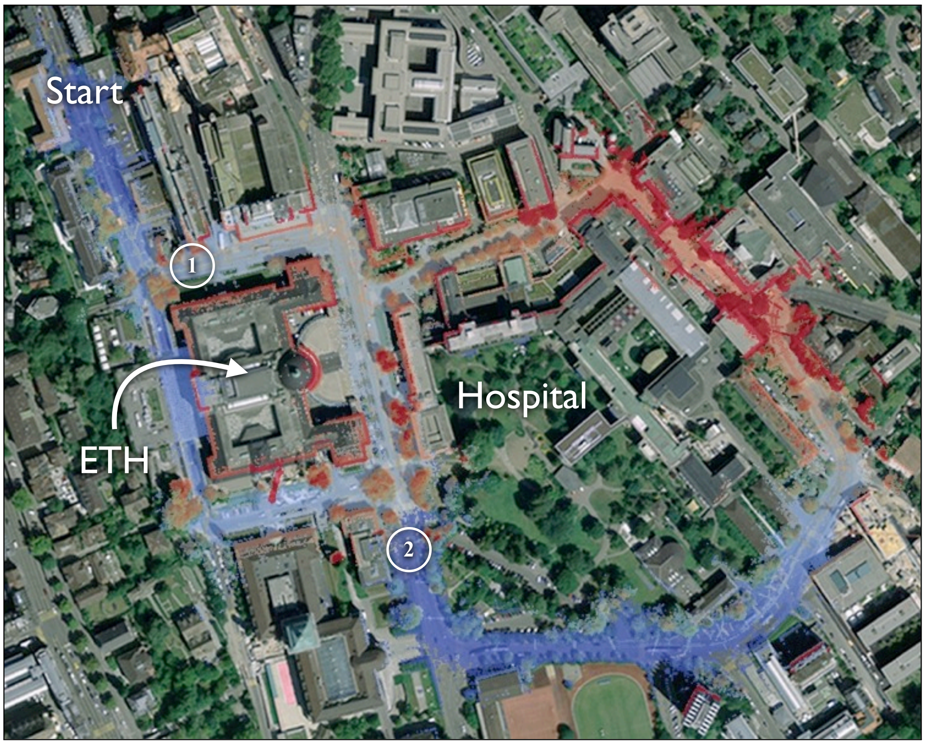


Figure 5.18: Overlay of a large scale reconstruction of the ETH main building and its surroundings with an aerial view. Colors represent the elevation, dark blue being low and dark red being high. Loop closing points are marked by numbers in circle, (1) being the small loop, (2) being the larger loop. The orthogonal projection of the aerial image was provided by the Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000

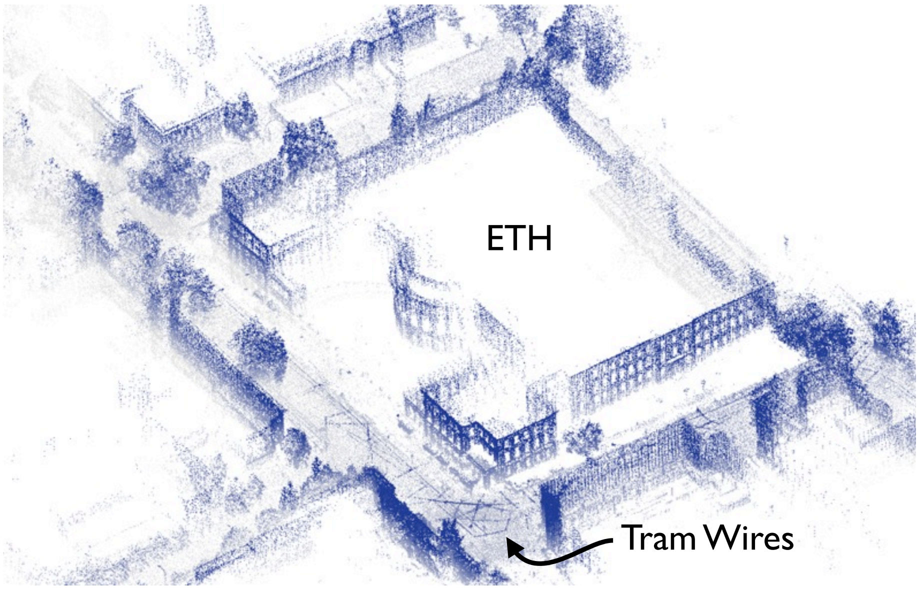


Figure 5.19: Bird view of the ETH main building. Colors represent elevation, light gray being low, dark blue being high.

5.5 Summary

This chapter covered a wide range of applications that were performed by different types of robotic platforms. The key characteristics of the robots employed in those applications are recapitulated in Table 5.7. We presented registration utilizations in situation awareness for Search & Rescue activities. The feasibility of real-time mapping deployments was demonstrated in a confined staircase, in open outdoor areas and in a collapsed church. We also demonstrated that surveying images recorded by an aerial vehicle and scans from ground vehicle can be gathered to enhance a scene reconstruction of a heavily damaged deployment site. A solution based on compact inspection systems was also suggested for mapping unreachable components in power plants. Such solutions could help reducing costs, time and dangers for the operators by bringing rapidly 3D informations from inside the part inspected, without the need for complex structures to support the operators. Finally, large-scale environmental surveys were shown to be successful without the need for specific loop closure algorithms, whether the survey is on water or the road.

Table 5.7: Summary of the robot key characteristics influencing the proposed registration solutions.

	Magnebike	*NiftiBot*	*Lizhbeth*	*Artor*	*SmartTer*
Weight (kg)	0.34	20	120	300	730
Volume (m^3)	0.006	0.17	0.96	6.75	6.38
Speed (m/s)	0.045	0.3	0.7	1.2	4.17
Depth Sensor	URG-04LX	LMS-151	HDL-32E	HDL-32E	LMS-291
Number of Points	340,000	55,000	50,000	45,000	28,000
Scan Rate (Hz)	0.02	0.35	1.6	5	1
Point Rate (Hz)	6,800	19,250	80,000	225,000	28,000

Based on the Search & Rescue deployments realized, we observed that most of the exploration activities are linear or expend following a start-shape. Most of the tasks imply: (1) going somewhere where no direct sight is possible from a safe zone, (2) assessing the situation and damages, and (3) backtracking the robot's path to the control station. Those task rarely imply loop closing and a coherent representation will be enough to bring the robot back to the control station even with some drift. Although none of the applications described in this chapter used loop closure, the street survey with the SmartTer would not be the appropriate solution in its current form. It is a good example of the utility of error relaxation and loop closing.

Having more accurate local registrations relaxes the pressure on loop closing, thus extending its reach to even larger loop.

Some lessons were learned while tuning the registration solutions for each one of the applications. One of the main parameter to tune first would be the maximal density required by the application. It can rapidly resolve real-time issues and remove local minima from the registration minimization. The maximal density and the maximum number of points in the global map depend on the expected size of the explored zone. For the specific case of exploration using NifitBot, the fact that the sensor is 0.21 m above the ground greatly reduces the motion planning range on a cluttered floor. Also, speed reduction should automatically be applied when the elements around the robot get too close, e.g., when crossing doorways. This would avoid the rapid decrease of the overlap between scans, which would drag down the registration quality. As for Artor, its fast motions required the handling of real-time processing in priority. During tuning, the fact that 3D scans jumped in critical situations (e.g., quick rotations) was enough to break the chain of registrations. In the case of inspection using the Magnebike, our evaluations provided better specifications for the construction of a new compact rotating laser. This new sensor will rotate faster (10 rpm) and will read from a UTM-30LX instead of the more noisy URG-04LX. This should bring the quality of inspection closer to the goal of 0.01 m precision that is required for defect detection. The preliminary work on the autonomous surface vessel, Lizhbeth, lead to a prioritization on the orientation estimation. For next applications, we decided to replace the Velodyne by a cheaper UTM-30LX moved by a custom tilting mechanism, allowing variable vertical scanning angles. This would give faster scanning rates when the obstacles are far away, while having the possibility to scan overhead vegetation when close to the shore. Those advantages come at the expense of range. Further tests will be needed to determine at which distance from the shore GPS navigation will give the hand to the localization based on laser. The projects implying the SmartTer finished a few years ago, but it was nevertheless interesting to push the capabilities of local registration to its limit. One of main observation is that even though the SICK LMS-291 specifications mentioned a maximal range of 70 m, the usable range remains under 20 m on concrete roads. Velodyne sensors can exceed this reach by three to four times, which can lower the cumulated error over large distances. Although more powerful, Velodyne sensors produce dense reading that are shaped like concentric disks, which can create local minima when registered. The use of past registered scan, in our case the global map, can help reducing those reconstruction artifacts and provides a more accurate reconstruction.

All examples demonstrate the added value of a modular ICP chains as each application has a specific set of requirements, that can still be fulfilled with the same open-source ICP library. The text-based parameter configurations combined with visual debugging tools allowed us to rapidly tune and understand limitations of configurations in order to achieved fast and accurate solutions.

Chapter 6
Conclusion

This thesis presented our work for a solid scientific methodology in robotics. At the beginning, we highlighted some flaws related to geometric registration evaluations, and acknowledged some difficulties encountered by researchers. The generic problem of registration for mobile robotics is a complex one with multiple facets. We focused on how to select, evaluate and develop 3D geometric registration algorithms in accordance to past publications. There are many challenges faced by researchers when dealing with registration applied to robotics. Notably, we exposed the large number of available publications and the lack of common grounds, which complicates the selection of an appropriate method. Also, given the diverse applications with non-uniform sensors, motion capabilities, environments, etc., it can be difficult to efficiently relate works between them. Finally, the lack of deeper solution characterizations and knowledge about algorithm limitations mislead researchers when implementing state-of-the-art solutions on a platform, costing time and resources. We took on those problems by four means: (1) an exhaustive literature review, (2) the elaboration of methods and tools specific for geometric registrations and applied to robotics, (3) a deeper investigation of well-known solutions, (4) an enumeration of the lessons learned based on concrete implementations on real-case scenarios.

6.1 Scientific Contributions

A detailed literature review was realized by surveying an extensive literature about registrations problems. By looking at the literature in related fields, we discovered that medical imaging reported their advancements in a more structured manner than in robotics. This may be because research with impacts on human requires deeper investigations before being applied in clinical tests. A generic structure emerged from this review, leading to a software framework. Although we covered a considerable amount of publications, many more remained unlisted in our survey. Further investigation is required to ensure that our proposed classification handles appropriately other scientific contributions on geometric registration. Moreover, the elaboration of a formal theoretical framework, either specialized for ICP or more generic to geometric registration would be the next step to conceal our proposed framework. Probabilistic methods, now providing more and more solutions based on entropy, would benefit from a stronger link with shape morphing to list possible solutions and their advantages.

Methods and tools were proposed to accelerate solution comparisons and to evaluate them against standard perturbations. Data sets with ground truth positioning, which was accurate enough to evaluate geometric registrations, were recorded in a variety of environments. The stable recording protocol followed for all data sets makes it a tool of choice for the transition between theoretical algorithms and field tests. In addition, we provided an evaluation protocol with predefined error metrics and controlled inputs that we hope will be accepted as standards in a short future. Based on the fact that most of the geometric registration algorithms share the same processing structure, we developed a modular library gathering ICP solutions in the same framework. We used this library to optimize solutions for real-time processing of high-frame rate sensors and to evaluate standard ICP solutions.

All specific problems tackled by researchers are unique, in a sense, and any evaluation framework will not fulfill all the specific needs. Nevertheless, it is the responsibility of the scientists to ensure an appropriate level of comparison, which necessarily implies a standardization of the constraints. One way to deal with this issue is to evaluate the proposed algorithm in a generic framework, and then use the analysis to motivate a more precise set of experiments. This procedure gives enough information to relate the works together, while drawing attention to specific conditions where the solution might be useful.

A concrete example would be someone proposing a registration solution that minimizes the scale in addition to a rigid transformation. The solution still needs to deal with different environments, variable overlaps of scans, and multiple types of motions. Moreover, the estimated scale should be close to identity when only rigid perturbations are applied. Once those robustness are demonstrated, specific experiments looking specifically at the scale can be analyzed. Of course, there is a limit in the assessment of an algorithm outside its intended conditions. This would be the case for registration solutions that heavily rely on labelled points based on colors. Although we specialized our effort on geometric registration, we believe that our contributions cover a large number of active research fields.

In Chapter 2, we saw that there are a multitude of solutions that still need to be tested. Our library currently only have few of those solutions implemented, and further involvement will be required to challenge more solutions.

Deeper investigations of basic solutions were realized to reveal some of their strengths, limitations and underlying assumptions. A suite of laser rangefinders is typically used in robotics, and some noise characterizations were still missing for some of them. We produced random noise models for the UTM-30LX, LMS-151 and reviewed older models proposed for the URG-04LX. Using the same experiment, we also defined lower bounds for surface reconstruction based on the noise produced by each sensors. The study presented results only for a random noise but static noise can also have a considerable impact on geometric registrations. Further evaluations are required to analyze the possibility of a calibration model that could also take into account the reflectivity, incidence angle and distance of a measured point.

We also analyzed the impact of environments, trajectories and initial perturbations on standard ICP methods using point-to-point and point-to-plane minimization scheme. Our main observation is that computation speed can be gained by fast reduction methods, without degrading the accuracy of the solution in an unusable range. The first set of standard evaluation results based on our proposed protocol were realized to demonstrate its viability and the range of analysis that can be rapidly done. What is still missing is an approbation of the scientific community for the proposed methods and a common repository to achieve the results. Hopefully, the transparency of open source softwares and publicly available results will help us to build a community, without refraining researchers for privacy or exclusivity concerns.

Lessons learned based on real scenarios were provided as a set of solutions using different parameters but sharing the same implementation. The same library, coupled with the analysis realized, allowed us to rapidly develop application prototypes that have the common criterion of respecting real-time constraints. The platforms used, Magnebike, NiftiBot, Lizhbeth, Artor and SmartTer, covered a variety of situations. Their different velocities, inertias, environments and locomotion types provided essential and unique information for for further implementations. Those study cases were also used to demonstrate that, even though its modularity, our library still provides strong computational performance. The results obtained help to readjust prior platform designs and demonstrated the viability of geometric registrations in a multitude of applications. Despite the fact that the transition from theory to implementation was done rapidly, more tests are required to ensure the stability of the prototypes presented.

6.2 Outside the Box

Robotics is a young research field compare to more traditional fields like medicine, biology, or geology. Therefore, it may be worthwhile to have a short glimpse on how those fields dealt with methodology problems. Focalizing on observations and the impacts of different parameters on a system is a clear concept in natural sciences, but in robotics, we often invest way more time to build the system than the time we devote to make observations with the platform. Although this might be true, other fields have similar long development phases as a major constraint. One good example is the research in medicine, or more precisely in pharmaceutics. The development of new medicaments are engineered based on more theoretical fields like chemistry, and must pass through a strict sequence of experimentations, from chemical stability, *in vitro* tests, to clinical evaluations. When it is not done properly, public health safety is endangered and harmful consequences may follow.

To achieve such deep investigations while working on a solution, time must be invested. In biology, studies are carried over multiple *PhD generations* to ensure that the full sequence of tests is done, and that safety is ensured at the end. This is rarely the case in robotics, which presents itself as a fast-paced field, where results from last year are already obsolete.

Sometimes overlooked, a robotic platform is an agent interacting with the environment. This link strongly defines the success or failure of a task, but the number of parameters influencing the environment can be overwhelming. Again, we can inspire ourself by what is seen in other fields to overcome the challenge. Field biologists face the same problematics and often opposed their conclusions based on observations made *in situ*, where the number of parameters is high, against the ones of colleagues who make most of their observations *in vitro*, i.e. under a controlled and limited set of parameters. We slowly see this distinction appearing in specialized journals and conferences on field deployments of robots, but it remains unclear what is a good field deployment and what validations are essential to go a step further and authorize the use of an autonomous systems in society.

Whenever there is a potential for strong and adverse consequences, researchers tend to follow a solid and structured methodology to avoid wrong interpretations. The open question is then: what will be those negative consequences for the public that would motivate the robotics researchers to apply a standardized methodology and how, as a community, we want to face them?

6.3 Impact on Society

This particular work might have a limited impact on society, but the robotics research in general does. Clearly, robots captivate the imagination and are always seen as what the future will be. Making the effort to provide tools to people that do not have thorough technical skills is also a good way to seek for new research opportunities and support creativity of others. This happened for photographers with image registrations, and it might also be the case for geometric registration. At its current stage, our library targets researchers and developers but with a bit of work, it could also support home made 3D games (e.g., for characters and scene modeling), amateur 3D movies, or even a new form of artistic expression...

Appendix – Transformation Functions

A frame of reference $\mathbb{F}$ in a 3D Euclidian space as been defined by an origin point $\mathbb{F}_o$ and a set $\mathbb{F}_{axis}$ of 3 orthogonal unit vectors, namely $\mathbb{F}_i, \mathbb{F}_j, \mathbb{F}_k$. Some parametrized shapes may have the same definition leading to confusion in the notion. Therefore, double letters (e.g. $\mathbb{A}, \mathbb{B}, \mathbb{C}$, ...) will be used when using frames as a convention. Coordinates correspond to a scale on each axis, which can be express as a triple $[x, y, z]^T$. The components of $\mathbb{F}$ can be explicitly defined in its own reference frame as:

$$\mathbb{F} = \{\mathbb{F}_o, \mathbb{F}_{axis}\}, \text{ with } \mathbb{F}_{axis} = \{\mathbb{F}_i, \mathbb{F}_j, \mathbb{F}_k\}, \text{ and}$$

$$\mathbb{F}_o = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \mathbb{F}_i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbb{F}_j = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbb{F}_k = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The notation ${}^{\mathbb{F}}\mathcal{S}$ will be used to explicit the fact that a shape S is express in a frame of reference $\mathbb{F}$.

The basic nomenclature and properties of transformation functions used in this work are defined here. A generic transformation function is written as follow:

$${}^{\mathbb{B}}p = {}^{\mathbb{B}}_{\mathbb{A}}\mathcal{T}(\Theta, {}^{\mathbb{A}}p),$$

which can be read as a transformation function $\mathcal{T}$ taking a primitive p expressed in a frame of reference $\mathbb{A}$ and transfering it to a new frame of reference $\mathbb{B}$. This transformation is defined by a set of parameters Θ. A transformation function is always written with 2 double letters, which can be read from $\mathbb{A}$ to $\mathbb{B}$. When the parameters are not of interest, a shorter notation is used:

$${}^{\mathbb{B}}p = {}^{\mathbb{B}}_{\mathbb{A}}\mathcal{T}\ {}^{\mathbb{A}}p$$

A chained application of transformation function:

$$^{\mathbb{C}}p = \, ^{\mathbb{A}}_{\mathbb{C}}\mathcal{T}(\Theta_3, \, ^{\mathbb{A}}p) = \, ^{\mathbb{C}}_{\mathbb{B}}\mathcal{T}\Big(\Theta_2, \, ^{\mathbb{B}}_{\mathbb{A}}\mathcal{T}(\Theta_1, \, ^{\mathbb{A}}p)\Big)$$

can be expressed in homogeneous coordinates as a matrix multiplications:

$$^{\mathbb{C}}p = \, ^{\mathbb{C}}_{\mathbb{A}}\mathcal{T} \, ^{\mathbb{A}}p = \, ^{\mathbb{C}}_{\mathbb{B}}\mathcal{T} \, ^{\mathbb{B}}_{\mathbb{A}}\mathcal{T} \, ^{\mathbb{A}}p$$

Changing the direction of transformation matrix is equivalent to use the inverse of the same matrix:

$$^{\mathbb{B}}_{\mathbb{A}}\mathcal{T} = \, ^{\mathbb{A}}_{\mathbb{B}}\mathcal{T}^{-1}$$

Transformation can be applied without parameter based on some conventions. To make the distinction, we used the term *mapping function* instead of transformation function. For example, the mapping function from Cartesian $\mathbb{C}$ to Spherical $\mathbb{S}$ coordinates is:

$$^{\mathbb{S}}_{\mathbb{C}}\mathcal{M}(x, y, z) = \begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2 + z^2} \\ \text{atan2}(y, x) \\ \text{asin}(\frac{z}{r}) \end{bmatrix}$$

The mapping function from Spherical $\mathbb{S}$ to Cartesian $\mathbb{C}$ coordinates is:

$$^{\mathbb{C}}_{\mathbb{S}}\mathcal{M}(r, \theta, \phi) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos\phi\cos\theta \\ r\cos\phi\sin\theta \\ r\sin\phi \end{bmatrix}$$

The mapping from Homogeneous $\mathbb{H}$ to Cartesian $\mathbb{C}$ coordinates is:

$$^{\mathbb{C}}p = \, ^{\mathbb{C}}_{\mathbb{H}}\mathcal{M}(\, ^{\mathbb{H}}p) = \, ^{\mathbb{C}}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \, ^{\mathbb{H}}\begin{bmatrix} x/w \\ y/w \\ z/w \end{bmatrix}$$

There is an infinite number of solutions to apply the mapping from Cartesian $\mathbb{C}$ to Homogeneous $\mathbb{H}$ coordinates, since it is augmenting the dimensionality. Although, to use homogeneous coordinates to linearly combined rotation and translation, the extra component w is assigned to one:

$$^{\mathbb{H}}_{\mathbb{C}}\mathcal{M}(x, y, z) = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The other group of transformation uses parameters to modify the space. Those transformation are: Translation, Uniforme Scaling, Rotation, Nonuniform Scaling, Shear, Vector Projection and Planar Projection; and there parameters are defined in Table A.1. A convent way to combine linearly transformation is the use their homogeneous representation as shown in Table A.2.

Table A.1: Basic transformation function minimal set of parameters. * note: there exist multiple way to express rotation: quaternions, Euler axis, Euler angle, rotation matrix. We only show two basic examples here.

Function	*Notation*	*Parameters*	*Note*
Translation	$T(t)$	$t = \{t_x, t_y, t_z\}$	
Uniforme Scaling	$U(s)$	s	
Rotation*	$R(r, \alpha)$	$r = \{r_x, r_y, r_z\}, \alpha$	Needs one line and an angle
Basic axis in $\mathbb{F}$	$R_x(\alpha)$	α	$= R(\mathbb{F}_x, \alpha)$
	$R_y(\alpha)$	α	$= R(\mathbb{F}_y, \alpha)$
	$R_z(\alpha)$	α	$= R(\mathbb{F}_z, \alpha)$
Yaw-Pitch-Roll	$R(\alpha)$	$\alpha = \{\alpha_x, \alpha_y, \alpha_z\}$	$= R_x(\alpha_x)R_y(\alpha_y)R_z(\alpha_z)$
Nonuniform Scaling	$S(s)$	$s = \{s_x, s_y, s_z\}$	
Shear	$H(h)$	$h = \{h_{xy}, h_{xz}, h_{yx}, h_{yz}, h_{zx}, h_{zy}\}$	
Vector Projection	$P(p)$	$p = \{p_x, p_y, p_z\}$	Needs one line
Planar Projection	$P(p_1, p_2)$	$p = \{p_x, p_y, p_z\}$	Needs two lines

Table A.2: Basic transformation function express in their homogeneous forms. * note: there exist multiple way to express rotation: quaternions, Euler axis, Euler angle, rotation matrix. We only show one basic example here.

Function	*Parameters*	*Homogeneous form*
Translation	t_x, t_y, t_z	$T(t) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Uniforme Scaling	s	$U(s) = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotation*	α, β, γ	$R_x(\alpha)R_y(\beta)R_z(\gamma) =$ $\begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta & 0 \\ c\alpha s\gamma + c\gamma s\beta s\alpha & c\gamma c\alpha - s\beta s\gamma s\alpha & -c\beta s\alpha & 0 \\ s\gamma s\alpha - c\gamma c\alpha s\beta & c\gamma s\alpha + c\alpha s\beta s\gamma & c\beta c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Nonuniform Scaling	s_x, s_y, s_z	$S(s) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Shear	h_{xy}, h_{xz} h_{yx}, h_{yz} h_{zx}, h_{zy}	$H(h) = \begin{bmatrix} 1 & h_{xy} & h_{xz} & 0 \\ h_{yx} & 1 & h_{yz} & 0 \\ h_{zx} & h_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Vector Projection	p_x, p_y, p_z	$P(p) = \begin{bmatrix} p_x & p_y & p_z & 0 \end{bmatrix}$
Planar Projection	p_x, p_y, p_z q_x, q_y, q_z	$P(p, q) = \begin{bmatrix} p_x & p_y & p_z & 0 \\ q_x & q_y & q_z & 0 \end{bmatrix}$

List of Figures

List of Tables

Bibliography

M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart. Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments. In *Robotics and Automation, 2011. Proceedings of the IEEE International Conference on*, pages 3056–3063, 2011. 84

M. Alwan, M. Wagner, G. Wasson, and P. Sheth. Characterization of Infrared Range-Finder PBS-03JN for 2-D Mapping. In *Robotics and Automation, 2005. Proceedings of the IEEE International Conference on*, pages 3936–3941, 2005. 68, 69

F. Amigoni, M. Reggiani, and V. Schiaffonati. An insightful comparison between experiments in mobile robotics and in science. *Autonomous Robots*, 27(4):313–325, Aug. 2009. 8, 35

L. Armesto, J. Minguez, and L. Montesano. A generalization of the metric-based Iterative Closest Point technique for 3D scan matching. In *Robotics and Automation, 2010. Proceedings of the IEEE International Conference on*, pages 1367–1372, 2010. 12, 15, 24, 25, 28, 31

S. Arya and D. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Discrete Algorithms, 1993. Proceedings of the 4th Annual ACM-SIAM Symposium on*, pages 271–280. Department of Computer Science, University of Maryland, College Park, Maryland, 20742 Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Mary- land, 20742, 1993. 26, 82, 86

G. Atanacio-Jimenez, J.-J. Gonzalez-Barbosa, J. B. Hurtado-Ramos, F. J. Ornelas-Rodriguez, H. Jimenez-Hernandez, T. Garcia-Ramirez, and R. Gonzalez-Barbosa. LIDAR Velodyne HDL-64E Calibration Using Pattern Planes. *International Journal of Advanced Robotic Systems*, 8(5): 70–82, 2011. 68

S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, pages 1–8, Oct. 2007. 51

R. Benjemaa and F. Schmitt. Fast global registration of 3D sampled surfaces using a multi-z-buffer technique. In *3-D Digital Imaging and Modeling, 1997. Proceedings of the International Conference on Recent Advances in*, pages 113–120, 1997. 26

R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(5):540–547, 1996. 30

P. Besl and H. McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, Feb. 1992. 5, 6, 30, 32, 67, 85

P. J. Besl. Geometric modeling and computer vision. In *Proceedings of the IEEE*, pages 936–958, 1988. 5, 19

P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *Intelligent Robots and Systems, 2003. Proceedings of the IEEE/RSJ International Conference on*, 2003. 32

M. Bosse and R. Zlot. Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM. *The International Journal of Robotics Research*, 27(6):667–691, June 2008. 12, 16, 24

M. Bosse and R. Zlot. Continuous 3D scan-matching with a spinning 2D laser. In *Robotics and Automation, 2009. Proceedings of the IEEE International Conference on*, pages 4312–4319, 2009a. 12, 16, 22, 23, 25, 29

M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2D lidar maps. *Robotics and Autonomous Systems*, 57(12):1211–1224, 2009b. 22, 23, 26, 30

M. Bosse, P. Newman, and J. L. S. Teller. Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *The International Journal of Robotics Research*, 23(12): 1113–1139, Dec. 2004. 8

K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Computer Vision And Image Understanding*, 101(1):1–15, Jan. 2006. 6, 11

A. Censi. An ICP variant using a point-to-line metric. In *Robotics and Automation, 2008. Proceedings of the IEEE International Conference on*, pages 19–25, 2008. 16, 22, 25, 26, 28, 30

G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3D object localization. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 83–89, 1992. 5, 6, 25, 30

Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings of the IEEE International Conference on*, pages 2724–2729. IEEE Comput. Soc. Press, Apr. 1991. 5, 6, 32, 36, 67, 79, 85

Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image And Vision Computing*, 10(3):145–155, Apr. 1992. 6, 30

D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The Trimmed Iterative Closest Point algorithm. In *Pattern Recognition, 2002. Proceedings of the 16th International Conference on*, pages 545–548, 2002. 28, 80, 85

S. Cousins, B. Gerkey, K. Conley, and W. Garage. Sharing Software with ROS. *Robotics & Automation Magazine, IEEE*, 17(2):12–14, 2010. 50

J. Diebel, K. Reutersward, S. Thrun, J. Davis, and R. Gupta. Simultaneous localization and mapping with active stereo vision. In *Intelligent Robots and Systems, 2004. Proceedings of the IEEE/RSJ International Conference on*, pages 3436–3443 vol.4, 2004. 14, 16, 24, 28, 29, 31

S. Druon, M. J. Aldon, and A. Crosnier. Color constrained ICP for registration of large unstructured 3d color data sets. In *Information Acquisition, 2006. Proceedings of the IEEE International Conference on*, pages 249–255. IEEE, 2006. 12, 15, 23, 24, 25, 28, 30

D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of CAD Models. *Computer Vision And Image Understanding*, 69(3):253–272, Feb. 1998. 8, 27

N. Fairfield and D. Wettergreen. Evidence grid-based methods for 3D map matching. In *Robotics and Automation, 2009. Proceedings of the IEEE International Conference on*, pages 1637–1642, May 2009. 22, 23

O. D. Faugeras and M. Hebert. The Representation, Recognition, and Locating of 3-D Objects. *The International Journal of Robotics Research*, 5(3):27–52, Sept. 1986. 4

J. Feldmar and N. Ayache. Locally affine registration of free-form surfaces. In *Computer Vision and Pattern Recognition, 1994. Proceedings of the IEEE Computer Society Conference on*, pages 496–501, 1994. 25

J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2): 99–119, May 1996. 15, 27, 30

M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 32

M. Fleischmann and S. Pons. Electrochemically induced nuclear fusion of deuterium. *Journal of Electroanalytical Chemistry and Interfacial Electrochemistry*, 261(2, Part 1):301 – 308, 1989. ISSN 0022-0728. doi: 10.1016/0022-0728(89)80006-3. 35

O. Fluck, C. Vetter, W. Wein, A. Kamen, B. Preim, and R. Westermann. A survey of medical image registration on graphics hardware. *Computer Methods and Programs in Biomedicine*, 104(3):e45–e57, Dec. 2011. 11

H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3-D model building. In *Computer Vision and Pattern Recognition, 1994. Proceedings of the IEEE Computer Society Conference on*, pages 581–586, 1994. 30

N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the ICP algorithm. In *3-D Digital Imaging and Modeling, 2003. Proceedings of the Fourth International Conference on*, pages 260–267, 2003. 27, 30, 81

D. Gingras, T. Lamarche, J. Bedwani, and E. Dupuis. Rough Terrain Reconstruction for Rover Motion Planning. In *Computer and Robot Vision, 2010. Proceedings of the Canadian Conference on*, pages 191–198, 2010. 81

G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Videometric III. Proceedings of SPIE Conference on*, pages 279–290, 1994. 16, 23, 24, 25, 26, 29, 30

S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, Aug. 1998. 26

G. Grisetti, C. Stachniss, and W. Burgard. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In *Robotics and Automation, 2005. Proceedings of the IEEE International Conference on*, pages 2432–2437, 2005. 16

G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *Intelligent Robots and Systems, 2007. Proceedings of the IEEE/RSJ International Conference on*, pages 3472–3478, 2007. 15

M. Hebert and E. Krotkov. 3-D measurements from imaging laser radars: how good are they? In *Workshop on Intelligence for Mechanical Systems, IROS 1991. IEEE/RSJ International*, pages 359–364, 1991. 68

P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, Apr. 2012. 8

G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch, J. Pernthaler, and R. Siegwart. Autonomous Inland Water Monitoring: Design and Application of a Surface Vessel. *Robotics & Automation Magazine, IEEE*, 19(1):62–72, Mar. 2012. 96, 115

S. Hrabar. An evaluation of stereo and laser-based range sensing for rotorcraft unmanned aerial vehicle obstacle avoidance. *Journal of Field Robotics*, 29 (2):215–239, Jan. 2012. 68

A. S. Huang, M. Antone, E. Olson, L. Fletcher, D. Moore, S. Teller, and J. Leonard. A High-rate, Heterogeneous Data Set From The DARPA Urban Challenge. *The International Journal of Robotics Research*, 29(13): 1595–1601, 2010. 36

J. R. Hurley and R. B. Cattell. The procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral Science*, 7(2): 258–262, Apr. 1962. 4

S. Jain, S. Nandy, G. Chakraborty, C. Kumar, R. Ray, and S. Shome. Error modeling of Laser Range Finder for robotic application using time domain technique. In *Signal Processing, Communications and Computing, 2011. Proceedings of the IEEE International Conference on*, pages 1–5, 2011. 68, 69

B. Jian and B. C. Vemuri. Robust Point Set Registration Using Gaussian Mixture Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1633–1645, 2011. 57

A. E. Johnson and S. B. Kang. Registration and integration of textured 3-D data. In *3-D Digital Imaging and Modeling, 1997. Proceedings of the International Conference on Recent Advances in*, pages 234–241, 1997. 25

T. Jost and H. Hügli. Fast ICP algorithms for shape registration. In *Pattern Recognition, 2002. Proceedings of the 24th DAGM German Symposium on*, pages 91–99. Pattern Recognition Group, Institute of Microtechnology, University of Neuchâtel, Breguet 2, CH-2000 Neuchâtel, Switzerland, 2002. 12, 16, 22, 25

T. Jost and H. Hugli. A multi-resolution scheme ICP algorithm for fast shape registration. In *3D Data Processing Visualization and Transmission, 2002. Proceedings of the First International Symposium on*, pages 540–543, 2002. 12, 16, 22, 23, 26

T. Jost and H. Hugli. A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images. In *3-D Digital Imaging and Modeling, 2003. Proceedings of the Fourth International Conference on*, pages 427–433, 2003. 27

T. Kahlmann, F. Remondino, and H. Ingensand. CALIBRATION FOR INCREASED ACCURACY OF THE RANGE IMAGING CAMERA SWISSRANGER. In *Image Engineering and Vision Metrology, 2006. Proceedings of the ISPRS Commission V Symposium on*, 2006. 68

K. Khoshelham and S. O. Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2):1437–1454, Feb. 2012. 68, 69, 72, 74, 141

D. Kim. A Fast ICP Algorithm for 3-D Human Body Motion Tracking. *Signal Processing Letters, IEEE*, 17(4):402–405, 2010. 14, 16, 25, 27, 30

L. Kneip, F. Tâche, G. Caprari, and R. Siegwart. Characterization of the compact Hokuyo URG-04LX 2D laser range scanner. In *Robotics and Automation, 2009. Proceedings of the IEEE International Conference on*, pages 1447–1454, 2009. 68, 69, 73, 74

G. J. M. Kruijff, M. Janicek, S. Keshavdas, B. Larochelle, H. Zender, N. J. J. M. Smets, T. Mioch, M. A. Neerincx, J. van Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlavac, T. Svoboda, T. Petricek, M. Reinstein, K. Zimmerman, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, H. Surmann, S. Corrao, S. Pratzler-Wanczura, and M. Sulk. Experience in System Design for Human-Robot Teaming in Urban Search and Rescue. In *Field and Service Robotics*, pages 1–14, Matsushima, Japan, July 2012. 96

P. Kumari, R. Shrestha, and B. Carter. Registration of LiDAR data through stable surface matching. In *Geoinformatics, 2009. Proceedings of the 17th International Conference on*, pages 1–5, 2009. 12

R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, 27(4):387–407, Sept. 2009. 56

P. Lamon, S. Kolski, and R. Siegwart. The SmartTer-a vehicle for fully autonomous navigation and mapping in outdoor environments. In *Proceedings of CLAWAR*, 2006. 120

K.-H. Lee and R. Ehsani. Comparison of two 2D laser scanners for sensing object distances, shapes, and surface patterns. *Computers and Electronics in Agriculture*, 60(2):250–262, Mar. 2008. 68, 69

C. Li, J. Xue, S. Y. Du, and N. Zheng. A Fast Multi-Resolution Iterative Closest Point Algorithm. In *Pattern Recognition, 2010. Proceedings of the Chinese Conference on*, pages 1–5, 2010. 27

X. Li, G. M. Xiong, Y. W. Hu, W. B. Li, Y. Jiang, J. W. Gong, and H. Y. Chen. Comparison of Single-Layer and Multi-Layer Laser Scanners for Measuring Characteristic. *Applied Mechanics and Materials*, 128-129: 548–552, Oct. 2011. 68, 69

Y. Liu. Automatic Range Image Registration in the Markov Chain. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):12–29, 2010. 12, 24, 26, 31

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 17, 25, 26, 28, 31, 32

F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, 1997. 7

M. Magnusson, A. Nüchter, C. Lorken, A. Lilienthal, and J. Hertzberg. Evaluation of 3D registration reliability and speed - A comparison of ICP and NDT. In *Robotics and Automation, 2009. Proceedings of the IEEE International Conference on*, pages 3907–3912, 2009. 16, 22, 23

M. Maimone, Y. Cheng, and L. Matthies. Two years of Visual Odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, 24(3):169–186, 2007. 38, 41

J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, Mar. 1998. 6, 8, 11

P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš. A review of 3D/2D registration methods for image-guided interventions. *Medical Image Analysis*, 16(3):642–661, Apr. 2012. 11

T. Masuda. Generation of geometric model by registration and integration of multiple range images. In *3-D Digital Imaging and Modeling, 2001. Proceedings of the Third International Conference on*, pages 254–261, 2001. 29

T. Masuda, K. Sakaue, and N. Yokoya. Registration and integration of multiple range images for 3-D model construction. In *Pattern Recognition, 1996. Proceedings of the 13th International Conference on*, pages 879–883 vol.1, 1996. 28

C. R. J. Maurer, G. B. Aboutanos, B. M. Dawant, R. J. Maciunas, and J. M. Fitzpatrick. Registration of 3-D images using weighted geometrical features. *Medical Imaging, IEEE Transactions on*, 15(6):836–849, 1996. 15

S. May, D. Droeschel, D. Holz, S. Fuchs, E. Malis, A. Nüchter, and J. Hertzberg. Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, 26(11-12):934–965, Nov. 2009. 56, 68

G. Medioni, C. K. Tang, and M. S. Lee. Tensor voting: Theory and applications. In *Reconnaissance des formes et Intelligence Artificielle, 2000. Proceedings of the Conference on*, 2000. 21, 32, 113

H. Mitasova, M. F. Overton, J. J. Recalde, D. J. Bernstein, and C. W. Freeman. Raster-Based Analysis of Coastal Terrain Dynamics from Multitemporal Lidar Data. *Journal of Coastal Research*, 252:507–514, Mar. 2009. 115

E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *Computer Vision and Pattern Recognition, 2005. Proceedings of the IEEE Computer Society Conference on*, pages 184–190 vol. 1, 2005. 25

A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an application in autonomous mine mapping. In *Robotics and Automation, 2004. Proceedings of the IEEE International Conference on*, pages 1998–2003 Vol.2, 2004. 8

A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM with approximate data association. In *Advanced Robotics, 2005. Proceedings of the 12th International Conference on*, pages 242–249, 2005. 26, 82

A. Nüchter, K. Lingemann, and J. Hertzberg. Cached k-d tree search for ICP algorithms. In *3-D Digital Imaging and Modeling, 2007. Proceeding of the Sixth International Conference on*, pages 419–426, 2007. 26

Y. Okubo, C. Ye, and J. Borenstein. Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation. In *Defense, Security, and Sensing, 2009. Proceedings of the SPIE Conference on*, 2009. 68, 69, 73, 74

Y. Pan, B. Dai, and Q. Peng. Fast and robust 3D face matching approach. In *Image Analysis and Signal Processing, 2010. Proceedings of the Second International Conference on*, pages 195–198, 2010. 12, 16, 23, 24, 25, 26, 29, 30, 31, 32

G. Pandey, J. R. McBride, and R. M. Eustice. Ford Campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13): 1543–1552, 2011. 36

C.-S. Park, D. Kim, B.-J. You, and S.-R. Oh. Characterization of the Hokuyo UBG-04LX-F01 2D laser rangefinder. In *Robot and Human Interactive Communication, 2010. Proceedings of the 19th IEEE International Symposium on*, pages 385–390, 2010. 68, 69

C.-S. Park, S.-W. Kim, D. Kim, and S.-R. Oh. Comparison of plane extraction performance using laser scanner and Kinect. In *Ubiquitous Robots and Ambient Intelligence, 2011. Proceedings of the 8th International Conference on*, pages 153–155, 2011. 69

J. Pascoal, L. Marques, and A. T. de Almeida. Assessment of Laser Range Finders in risky environments. In *Intelligent Robots and Systems, 2008. Proceedings of the IEEE/RSJ International Conference on*, pages 3533–3538, Nice, 2008. IEEE. 68, 69

K. Pathak, A. Birk, N. Vaškevičius, and J. Poppinga. Fast Registration Based on Noisy Planes With Unknown Correspondences for 3-D Mapping. *Robotics, IEEE Transactions on*, 26(3):424–441, 2010a. 36

K. Pathak, D. Borrmann, J. Elseberg, N. Vaskevicius, A. Birk, and A. Nüchter. Evaluation of the robustness of planar-patches based 3D-registration using marker-based ground-truth in an outdoor urban scenario. In *Intelligent Robots and Systems, 2010. Proceedings of the IEEE/RSJ International Conference on*, pages 5725–5730, 2010b. 54, 77

G. Paul, D. Liu, N. Kirchner, and G. Dissanayake. An effective exploration approach to simultaneous mapping and surface material-type identification of complex three-dimensional environments. *Journal of Field Robotics*, 26 (11-12):915–933, 2009. 68

T. Peynot, S. Scheding, and S. Terho. The Marulan Data Sets: Multi-Sensor Perception in Natural Environment With Challenging Conditions. *The International Journal of Robotics Research*, 29(13):1602–1607, Nov. 2010. 36

S. T. Pfister, K. L. Kriechbaum, S. I. Roumeliotis, and J. W. Burdick. Weighted range sensor matching algorithms for mobile robot displacement estimation. In *Robotics and Automation, 2002. Proceedings of the IEEE International Conference on*, 2002. 7

J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8):986–1004, 2003. 11, 31

F. Pomerleau, F. Colas, F. Ferland, and F. Michaud. Relative motion threshold for rejection in ICP registration. *Field and Service Robotics*, pages 229–238, 2010. 28

F. Pomerleau, B. Lescot, F. Colas, and M. Liu. Dataset Acquisitions for USAR Environments. In *Robot-Human Teamwork in Dynamic Adverse Environment, 2010. Proceedings of the AAAI Fall Symposium on*, 2011a. 35

F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart. Tracking a depth camera: Parameter exploration for fast ICP. In *Intelligent Robots and Systems, 2011. Proceedings of the IEEE/RSJ International Conference on*, pages 3824–3829. Autonomous Systems Lab. – ETH Zurich, Tannenstr 3, 8092 Zurich, Switzerland, 2011b. 35, 37, 63, 67

F. Pomerleau, A. Breitenmoser, M. Liu, F. Colas, and R. Siegwart. Noise Characterization of Depth Sensors for Surface Inspections. In *Applied Robotics for the Power Industry, 2012. Proceedings of the Second International Conference on*, pages 1–6, Zurich, Switzerland, Aug. 2012a. 67

F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging Data Sets for Point Cloud Registration Algorithms. *The International Journal of Robotics Research*, Sept. 2012b. 35

F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, 34(3):133–148, Feb. 2013. 35, 67, 96

K. Pulli. Multiview registration for large data sets. In *3-D Digital Imaging and Modeling, 1999. Proceedings of the Second International Conference on*, pages 160–168, Oct. 1999. 12, 16, 22, 25, 26, 28, 30

L. Reyes, G. Medioni, and E. Bayro. Registration of 3D points using geometric algebra and tensor voting. *International Journal of Computer Vision*, 2007. 32

J. Rudan, Z. Tuza, and G. Szederkenyi. Using LMS-100 laser rangefinder for indoor metric map building. In *Industrial Electronics, 2010. Proceedings of the IEEE International Symposium on*, pages 525–530. IEEE, 2010. 68

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings of the Third International Conference on*, pages 145–152, 2001. 7, 50, 67, 93

J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image And Vision Computing*, 2007. 11

R. Sanz-Cortiella, J. Llorens-Calveras, J. R. Rosell-Polo, E. Gregorio-Lopez, and J. Palacin-Roca. Characterisation of the LMS200 Laser Beam under the Influence of Blockage Surfaces. Influence on 3D Scanning of Tree Orchards. *Sensors*, 11(3):2751–2772, Mar. 2011. 68, 69

D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42, Mar. 2002. 51

W. Schroeder, K. Martin, and B. Lorensen. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, 4th edition, Dec. 2006. 63

C. Schutz, T. Jost, and H. Hugli. Multi-feature matching algorithm for free-form 3D surface registration. In *Pattern Recognition, 1998. Proceedings of the 14th International Conference on*, pages 982–984 vol.2, 1998. 16, 22, 23, 25

A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems V. Proceedings of the Conference on*, page 21. Stanford University, 2009. 12, 22, 25, 28, 30

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Computer Vision and Pattern Recognition, 2006. Proceedings of the IEEE Computer Society Conference on*, pages 519–528, 2006. 51

L. Silva, O. Bellon, and K. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5): 762–776, 2005. 31, 32

M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The New College Vision and Laser Data Set. *The International Journal of Robotics Research*, 28(5):595–599, May 2009. 36

C. Stewart, C.-L. Tsai, and B. Roysam. The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *Medical Imaging, IEEE Transactions on*, 22(11):1379–1394, 2003. 14, 16, 22, 23, 24, 25, 26, 28, 29, 31

E. Stumm, A. Breitenmoser, F. Pomerleau, C. Pradalier, and R. Siegwart. Tensor-voting-based navigation for robotic inspection of 3D surfaces using lidar point clouds. *The International Journal of Robotics Research*, 31 (12):1465–1488, Nov. 2012. 96, 112

F. Tâche, W. Fischer, G. Caprari, R. Siegwart, R. Moser, and F. Mondada. Magnebike: A magnetic wheeled robot with high mobility for inspecting complex-shaped structures. *Journal of Field Robotics*, 26(5), May 2009. 110

F. Tâche, F. Pomerleau, W. Fischer, G. Caprari, F. Mondada, R. Moser, and R. Siegwart. MagneBike: Compact magnetic wheeled robot for power plant inspection. In *Applied Robotics for the Power Industry, 2010. Proceedings of the First International Conference on*, pages 1–2, 2010. 96

F. Tâche, F. Pomerleau, G. Caprari, R. Siegwart, M. Bosse, and R. Moser. Three-Dimensional Localization for the MagneBike Inspection Robot. *Journal of Field Robotics*, 28(2):180–203, 2011. 96, 111

S. Thrun, W. Burgard, and D. Fox. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning*, 31(1/3): 29–53, 1998. 7

C. H. Tong and T. Barfoot. A self-calibrating 3D ground-truth localization system using retroreflective landmarks. In *Robotics and Automation, 2011. Proceedings of the IEEE International Conference on*, pages 3601–3606, 2011. 37

V. Tretyakov and T. Linder. Range sensors evaluation under smoky conditions for robotics applications. In *Safety, Security, and Rescue Robotics, 2011. Proceedings of the IEEE International Symposium on*, pages 215–220, 2011. 68, 69

C. Tsai, C. Li, G. Yang, and K. Lin. The Edge-Driven Dual-Bootstrap Iterative Closest Point Algorithm for Registration of Multimodal Fluorescein Angiogram Sequence. *Medical Imaging, IEEE Transactions on*, 29(3): 636–649, 2010. 14, 16, 22, 23, 25

T. Tuytelaars and K. Mikolajczyk. A survey on local invariant features. *Foundations and Trends in Computer Graphics and Vision*, 2008. 23

M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3): 358–367, Nov. 1991. 5

U. Wong, A. Morris, C. Lea, J. Lee, C. Whittaker, B. Garney, and R. Whittaker. Comparative evaluation of range sensing technologies for underground void modeling. In *Intelligent Robots and Systems, 2011. Proceedings of the IEEE/RSJ International Conference on*, pages 3816–3823, 2011. 68

O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. Benchmarking urban six-degree-of-freedom simultaneous localization and mapping. *Journal of Field Robotics*, 25(3):148–163, 2008. 27, 95

K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems. In *Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, 2010 (ICRA)*, 2010. 23

C. Ye and J. Borenstein. Characterization of a 2D laser scanner for mobile robot obstacle negotiation. In *Robotics and Automation, 2002. Proceedings of the IEEE International Conference on*, pages 2512–2518, Washington, 2002. IEEE. 68, 69

H. Yoshitaka, K. Hirohiko, O. Akihisa, and Y. Shin'ichi. Mobile Robot Localization and Mapping by Scan Matching using Laser Reflection Intensity of the SOKUIKI Sensor. In *Industrial Electronics, 2006. Proceedings of the IEEE 32nd Annual Conference on*, pages 3018–3023, 2006. 12, 16, 23, 24, 25

Z. Zhang. Point matching for registration of free-form surfaces. *Lecture Notes in Computer Science*, 719:460–467, May 1993. 6

Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, Oct. 1994. 6, 16, 28

Y. Zhuo and X. Du. Automatic registration of partial overlap three-dimensional surfaces. In *Mechanic Automation and Control Engineering, 2010. Proceedings of the International Conference on*, pages 299–302, 2010. 77

T. Zinsser, J. Schmidt, and H. Niemann. A refined ICP algorithm for robust 3-D correspondence estimation. In *Image Processing, 2003. Proceedings of the IEEE International Conference on*, pages II–695–8 vol.3, 2003. 28, 31

B. Zitová and J. Flusser. Image registration methods: a survey. *Image And Vision Computing*, 21(11):977–1000, 2003. 14, 30

R. Zlot and M. Bosse. Place recognition using keypoint similarities in 2D lidar maps. *Experimental Robotics*, 54:363–372, 2009. 26, 82

About the Author

François Pomerleau was born in a small town of Canada called Rivière-du-Loup, on the 20th of November 1981. He conduced is undergraduated studies in *la Belle Province de Québec* during which he work for the Canadian Space Agency and realized his early research work in collaboration with the European Space Agency. Those implications allowed him to develop and test a micro-satellite during a parabolic flight campaign. He received his M.Sc degree from the University of Sherbrooke on the topic of registration algorithms for tele-assistive home care robotics, under the supervision of Professor François Michaud.

During his Bachelor, he went for the first time in Lausanne, Switzerland, to work on the project SmartTer as an exchange student. Under the supervision of Professor Roland Siegwart, he designed a control system for an autonomous car and was hired as an engineer to support field deployments. In January 2009, he started is doctoral thesis at the Autonomous Systems Lab in ETH Zurich. The research realized implied a rich game of robots and was related to European project Robots@home (FP6-IST-045350) and NIFTi (FP7-ICT-47870). His research interests include 3D localization and mapping, registration algorithms and field robotics aiming at bringing robots outside research laboratories. He actively seek new opportunities to link his two passions: photography and robotics.

The reader is warmly welcome to submit questions, remarks and comments directly to the author (email: `f.pomerleau@gmail.com`).

List of Publications

Journals

- F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP Variants on Real-World Data Sets,” Autonomous Robots, vol. 34, no. 3, pp. 133-148, Feb. 2013.
- M.-È. Garneau, T. Posch, G. Hitz, F. Pomerleau, C. Pradalier, R. Siegwart and J. Pernthaler, “Short-term displacement of *Planktothrix rubescens* (cyanobacteria) in a pre-alpine lake observed using an autonomous sampling platform,” Limnology & Oceanography, *Accepted for publication, April 2013.*
- P. Krüsi, B. Bücheler, F. Pomerleau, P. Furgale, U. Schwesinger, R. Siegwart, “Lighting-Invariant Autonomous Route Following Using ICP,” Journal of Field Robotics, *Submitted June 2013.*
- V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, M. Reinštein, “Robust Data Fusion of Multi-modal Sensory Information for Mobile Robots,” Journal of Field Robotics, *Submitted June 2013.*
- F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, “Challenging data sets for point cloud registration algorithms,” The International Journal of Robotics Research, vol. 31, no. 14, pp. 1705-1711, Dec. 2012.
- E. Stumm, A. Breitenmoser, F. Pomerleau, C. Pradalier, and R. Siegwart, “Tensor-voting-based navigation for robotic inspection of 3D surfaces using lidar point clouds,” The International Journal of Robotics Research, vol. 31, no. 12, pp. 1465-1488, Nov. 2012.
- G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch, J. Pernthaler, and R. Siegwart, “Autonomous Inland Water Monitoring: Design and Application of a Surface Vessel,” Robotics & Automation Magazine, IEEE, vol. 19, no. 1, pp. 62-72, Mar. 2012.
- F. Tâche, F. Pomerleau, G. Caprari, R. Siegwart, M. Bosse, and R. Moser, “Three-Dimensional Localization for the MagneBike Inspection Robot,” Journal of Field Robotics, vol. 28, no. 2, pp. 180-203, 2011.
- F. Michaud, P. Boissy, D. Labonté, S. Brière, K. Perreault, H. Corriveau, A. Grant, M. Lauria, R. Cloutier, M. A. Roux, D. Iannuzzi, M. P. Royer, F. Ferland, F. Pomerleau, and D. Létourneau, “Exploratory design and evaluation of a homecare teleassistive mobile robotic system,” Mechatronics, vol. 20, no. 7, pp. 751-766, Oct. 2010.

Peer-reviewed Proceedings

- F. Pomerleau, A. Breitenmoser, M. Liu, F. Colas, and R. Siegwart, “Noise Characterization of Depth Sensors for Surface Inspections,” presented at the Applied Robotics for the Power Industry, 2012. Proceedings of the Second International Conference on, Zurich, Switzerland, 2012, pp. 1-6.
- G. J. M. Kruijff, M. Janicek, S. Keshavdas, B. Larochelle, H. Zender, N. J. J. M. Smets, T. Mioch, M. A. Neerincx, J. van Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlavac, T. Svoboda, T. Petricek, M. Reinstein, K. Zimmerman, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, H. Surmann, S. Corrao, S. Pratzler-Wanczura, and M. Sulk, “Experience in System Design for Human-Robot Teaming in Urban Search and Rescue,” presented at the Field and Service Robotics, Matsushima, Japan, 2012, pp. 1-14.
- S. Gemme, D. Gingras, A. Salerno, E. Dupuis, F. Pomerleau, and F. Michaud, “Pose Refinement Using ICP Applied to 3-D LIDAR Data for Exploration Rovers,” presented at the Artificial Intelligence, Robotics and Automation in Space, Proceedings of the International Symposium on, Turin, Italy, 2012
- F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, “Tracking a depth camera: Parameter exploration for fast ICP,” presented at the Intelligent Robots and Systems, 2011. Proceedings of the IEEE/RSJ International Conference on, 2011, pp. 3824-3829.
- F. Pomerleau, B. Lescot, F. Colas, and M. Liu, “Dataset Acquisitions for USAR Environments,” presented at the Robot-Human Teamwork in Dynamic Adverse Environment, 2010. Proceedings of the AAAI Fall Symposium on, 2011.
- F. Pomerleau, F. Colas, F. Ferland, and F. Michaud, “Relative motion threshold for rejection in ICP registration,” presented at the Field and Service Robotics. Proceedings of the International Conference on, 2010, pp. 229-238.
- F. Tâche, F. Pomerleau, W. Fischer, G. Caprari, F. Mondada, R. Moser, and R. Siegwart, “MagneBike: Compact magnetic wheeled robot for power plant inspection,” presented at the International Conference on Applied Robotics for the Power Industry. Proceedings of, 2010. Proceedings of the First International Conference on, 2010, pp. 1-2.
- M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, “Scale-only visual homing from an omnidirectional camera,” presented at the Robotics and Automation, 2012. Proceedings of the IEEE International Conference on, pp. 3944-3949.
- F. Ferland, F. Pomerleau, C. T. Le Dinh, and F. Michaud, “Egocentric and exocentric teleoperation interface using real-time, 3D video projection,” presented at the Human-Robot Interaction, Proceedings of the ACM/IEEE international conference on, 2009, pp. 37-44.